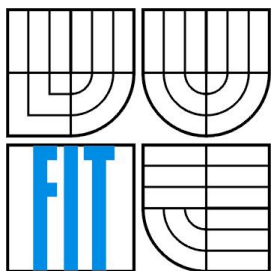


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SUDOKU NA MOBILNÍM TELEFONU

SUDOKU FOR MOBILE PHONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ondřej Pumprla

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Igor Szöke

BRNO 2007

Zadání bakalářské práce

Řešitel: **Pumpřla Ondřej**

Obor: Informační technologie

Téma: **Sudoku na mobilním telefonu**

Kategorie: Umělá inteligence

Pokyny:

1. Seznamte se s vývojovým prostředím Java pro mobilní telefony.
2. Seznamte se s pravidly hry Sudoku, jejího řešení a generování hracích polí.
3. Implementujte Sudoku do mobilního telefonu.
4. Implementujte aplikaci, která bude Sudoku řešit.
5. Zhodnoťte dosažené výsledky a navrhněte směry případného dalšího vývoje.

Literatura:

- Dle pokynu vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a alespoň část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

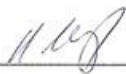
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing., UPGM FIT VUT**

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Příkopova 2



doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Ondřej Pumpřla**
Id studenta: 84174
Bytem: Sadová 1530, 769 01 Holešov
Narozen: 21. 08. 1984, Přílepy
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Sudoku na mobilním telefonu
Vedoucí/školicel VŠKP: Szöke Igor, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ☒ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

Tato práce se zabývá vývojem mobilní aplikace založené na programovacím jazyce Java - J2ME. Aplikací je hra Sudoku, přičemž zadání si uživatel může vygenerovat přímo v samotné mobilní aplikaci, nebo importovat z webové aplikace. Webová aplikace je naprogramována prostřednictvím PHP a MySQL. Generátor produkuje zadání, která jsou řešitelná na základě technik, které lidé běžně používají.

Klíčová slova

J2ME, CLDC, KVM, MIDP, RMS, MIDlet, HTTP, J2ME Wireless Toolkit, generická mobilní aplikace, algoritmizace Sudoku, Knuthův algoritmus X

Abstract

This thesis deals with development of application for mobile phones programmed in Java - J2ME. The application is a game Sudoku. User can solve automatically or manually generate Sudoku puzzle or import a Sudoku from a web application. The web application is programmed by using PHP and MySQL. The generator makes Sudoku puzzles which are solvable on basis of common used human-solvable techniques.

Keywords

J2ME, CLDC, KVM, MIDP, RMS, MIDlet, HTTP, J2ME Wireless Toolkit, generic mobile application, algorithm development Sudoku, Knuth's Algorithm X

Citace

Ondřej Pumprla: Sudoku na mobilním telefonu, bakalářská práce, Brno, FIT VUT v Brně, 2007

Sudoku na mobilním telefonu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Igora Szökeho.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Pumprla
7. května 2007

Poděkování

Děkuji svému vedoucímu Ing. Igoru Szökemu za cenné rady a konstruktivní připomínky při vývoji aplikace. Děkuji i svým rodičům za psychickou podporu při realizaci práce.

© Ondřej Pumprla, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
Úvod	3
1 Technologie J2ME	4
1.1 Obecný pohled na J2ME	4
1.2 Konfigurace	5
1.2.1 CDC	5
1.2.2 CLDC	6
1.3 Profily	6
1.3.1 MIDP 1.0	7
1.3.2 MIDP 2.0	8
1.4 MIDlet	8
2 Algoritmizace Sudoku	10
2.1 Obecné vlastnosti Sudoku	10
2.2 Řešení Sudoku	11
2.2.1 Řešení hrubou silou	11
2.2.2 Řešení s využitím heuristik	12
2.2.3 Knuthův algoritmus X	14
2.3 Generování Sudoku zadání	15
2.3.1 Využití existujících Sudoku zadání	15
2.3.2 Generování s využitím algoritmu pro řešení Sudoku	16
2.3.2.1 Postupné odkrývání čísel	16
2.3.2.2 Postupné přidávání nových čísel	16
3 Midlet Sudoku	18
3.1 Předběžné aspekty před tvorbou aplikace	18
3.2 Spuštění aplikace a úvodní menu	19
3.2.1 Generování nové hry Sudoku	20
3.2.1.1 Generování zadání	20
3.2.1.2 Uživatelské rozhraní	22
3.2.2 Automatický řešitel Sudoku zadání	24
3.2.3 Ukládání stavu rozehrané hry	25
3.2.4 Import zadání z webového serveru	25
3.2.5 Seznam časů úspěšných řešitelů	26
3.2.6 Nastavení, Nápověda, Autor	27

4	Webová aplikace.....	28
4.1	Účel webové aplikace.....	28
4.2	Způsob generování nových zadání.....	28
5	Porovnání s cizími aplikacemi.....	29
6	Závěr	31
	Literatura	32
	Seznam příloh	34

Úvod

Je doba, kdy si člověk jen stěží dokáže představit praktickou existenci bez mobilního telefonu. V současnosti jsou na trhu v drtivé většině mobilní telefony s podporou aplikační platformy J2ME (Java 2, Micro Edition). A právě v tomto programovacím prostředí existuje nespočetné množství programů, které uživatelům mobilních telefonů usnadňují práci, či zpříjemňují volné chvíle zábavou.

Bakalářská práce je zaměřena na oblast určenou pro zábavu, a to vytvoření hry Sudoku, jenž v poslední době dospívá velmi velké oblibě. Ačkoliv je zaměření aplikace určené pro zábavu, byla snaha využít co nejširší škálu platformy J2ME. Implementace aplikace byla však ubírána způsobem, který zajišťuje bezproblémovou přenositelnost na nejširší okruh mobilních telefonů s podporou Javy. Toto se mnohdy jeví jako nelehký úkol plynoucí například z diametrálně rozdílných rozměrů displeje, či velikosti paměti pro běhové prostředí Javy.

Aplikace má produkovat taková zadání, aby uživatel při řešení byl schopen dosáhnout cíle takovým způsobem, který by byl kompetentní k dané obtížnosti – tedy při lehčích zadání použít obecně známé lehčí metody a naopak při těžší obtížnosti použít těžší metody.

Téměř jakákoliv hra, ve které uživatel může dosáhnout bodového ohodnocení, obsahuje uchovávání výsledků a možnosti zobrazení seznamu těchto dosažených výsledků. Ukládání výsledků v mobilním telefonu však nese omezení v možnosti porovnat se s ostatními uživateli. Z tohoto důvodu je na místě rozšířit vlastnost aplikace o možnost nahrát si Sudoku zadání z webového serveru do mobilního telefonu. Jakmile uživatel dané zadání vyřeší, odesílá se zpět na webový server čas řešení a jméno autora tohoto času. Na serveru se data kumulují a uživatel se může lehce porovnat s ostatními hráči Sudoku aplikace. Webová aplikace je napsána v jazycích PHP a MySQL, komunikace probíhá prostřednictvím protokolu HTTP.

Aplikace je určená pro mobilní telefony s podporou Javy (J2ME) s barevným displejem, CLDC 1.0 konfigurací a MIDP 1.0 profilem. Čímž je zaručena větší přenositelnost.

Aplikace Sudoku byla v průběhu vývoje testována/emulována v Java 2 MicroEdition Wireless Toolkit, jenž je zdarma k dostání na oficiálních stránkách Javy [6]. V praktickém nasazení byla aplikace testována převážně na mobilním telefonu Sony Ericsson K750i.

Jedná se o samostatný projekt bez návaznosti na předešlou cizí práci.

V následujícím textu se bakalářská práce zabývá technologií J2ME, a to hlavně obecným popisem a souvislostmi s vývojem konkrétních mobilních aplikací. Další kapitola popisuje algoritmický přístup generování nových Sudoku matic a automatického řešení Sudoku zadání. Následující kapitola podrobněji popisuje implementaci některých vlastností mobilní aplikace Sudoku a jakými technologiemi bylo tohoto dosaženo. Čtvrtá kapitola pojednává o webové aplikaci, která slouží pro importování zadání a zobrazení výsledků. Poslední kapitolou je srovnání aplikace Sudoku v rámci bakalářské práce s jinými (komerčními a nekomerčními) aplikacemi.

1 Technologie J2ME

Jelikož byla mobilní aplikace vyvíjena v programovacím jazyce Java - J2ME, je vhodné uvést důležité vlastnosti tohoto jazyka, kterých programátor při tvorbě aplikací určených pro mobilní telefony využívá. Tato kapitola tedy popisuje vlastnosti jazyka Java - edice J2ME.

1.1 Obecný pohled na J2ME

Sun Microsystems definuje tři platformy jazyka Java, jenž odrážejí různé požadavky na výpočetní prostředí cílového systému, a to:

- Java 2 Standard Edition (J2SE)
- Java 2 Enterprise Edition (J2EE)
- Java 2 Micro Edition (J2ME)

Vznik platformy J2ME se uskutečnil (pod záštitou projektu *Java Community Process*) na požadavek vyvíjet aplikace pro zařízení, která by nezvládla chod aplikací založených na standardní edici (J2SE). Jedná se o jednu z nejrozšířenějších aplikačních platform pro elektronická zařízení, jako jsou např. mobilní telefony, osobní digitální asistenti (PDA), organizéři, pagery, navigační přístroje pro automobily a další malá zařízení.

Ačkoliv se obecně Java snaží o všeobecnou přenositelnost na různé cílové stanice, u edice J2ME se jedná o nelehký úkol. Tato obtíž realizace jednoznačné přenositelnosti je implikována širokou škálou účelnosti mobilních zařízení, které dominují variantními vlastnostmi. Například přídatná zařízení určená pro automobilový průmysl jsou obecně komplexnější než mobilní telefony. Relativně značné rozdíly se vyskytují i u zařízení primárně určených pro stejný účel, například low-end a high-end mobilní telefony, které se liší taktovací frekvencí procesoru, zobrazovacími schopnostmi, velikostí paměti pro běhové prostředí Javy nebo prostředky pro síťovou komunikaci. Z tohoto účelu se J2ME nejeví jako jediná obecná specifikace (neexistuje nejmenší společný jmenovatel), ale jako množina specifikací založená na *konfiguracích* a *profilech*.

Z obecného pohledu J2ME se centrem stává virtuální stroj (což je pro Javu typické), který pracuje na hostitelském operačním systému zařízení. Jedná se o skutečný javovský virtuální stroj podle specifikace pro virtuální stroje Javy, až na některé specifické odchylky, které vyplývají z použitelnosti na zařízeních s omezenými zdroji (převážně malá paměť). Nad virtuálním strojem je specifická konfigurace (viz. kapitola 1.2 *Konfigurace*) J2ME, která je ovlivněna dostupným množstvím paměti a výkonem procesoru. Vrchol konfigurace tvoří jeden či více profilů (viz. kapitola 1.3 *Profily*), využívající příbuzných funkcí v podobných zařízeních. [1]

Virtuální stroj, konfigurace a profily tvoří tzv. *pracovní prostředí J2ME*. Pracovní prostředí je obecně různé pro různé typy elektronických zařízení, což definuje různé možnosti použitelnosti programových prostředků. Na následujícím *obrázku 1.1* je souhrnně zobrazen vztah mezi jednotlivými částmi pracovního prostředí J2ME.



Obrázek 1.1: Přehled architektury pracovního prostředí J2ME

1.2 Konfigurace

Konfigurace definuje množinu základních programových knihoven, jež jsou ovlivněny technickými zdroji (velikost ROM a RAM paměti, typ a frekvence procesoru, zdroj napájení, typ síťového propojení) daného zařízení. Tedy zařízení s podobnými technickým vlastnostmi disponují stejnou konfigurací.

Konfigurace specifikuje tři základní elementy [2]:

- Množinu podporovaných rysů programovacího jazyka Java
- Vlastnosti virtuálního stroje Javy
- Podporované javovské knihovny a aplikační programová rozhraní (API)

V současné době existují dva typy konfigurací. Jedna je určena pro nízkoúrovňová zařízení, kde spadá většina mobilních telefonů. Této konfiguraci se říká CLDC (Connected Limited Device Configuration). Druhá konfigurace se jmenuje CDC (Connected Device Configuration) a je určena pro inteligentnější zařízení, jako jsou například PDA, navigační zařízení pro automobily, apod.

Při tvorbě aplikace Sudoku byla uvažována konfigurace CLDC, jelikož se jedná o konfiguraci, která se v drtivé většině používá u mobilních telefonů.

1.2.1 CDC

Konfigurace CDC se vyskytuje u výkonnějších zařízení (inteligentní PDA, chytré komunikátory, digitální Set-Top-Boxy). CDC podporuje všechny vlastnosti virtuálního stroje Java 2, a proto je zde k dispozici většina vlastností knihovny J2SE.

Virtuální stroj pro konfiguraci CDC se nazývá CVM (Compact Virtual Machine). Pro programátora aplikací určených pro mobilní telefony nemá v současné době velký význam.

Požadavky na zdroje pro CDC jsou následující [1]:

- Zařízení je řízeno 32-bitovým procesorem
- Pro běhové prostředí Javy zařízení disponuje 2 MB RAM a 2.5 MB ROM.
- Zařízení je připojitelné k některému typu sítě.
- Plná podpora virtuálního stroje Java 2

1.2.2 CLDC

Konfigurace CLDC je z pohledu programátora mobilních aplikací zajímavější, jelikož se vyskytuje v mobilních telefonech. Dále může být nalezena například v pagerech a starších PDA zařízení. Konfigurace CLDC je tedy zaměřena na zařízení, jež jsou výkonnostně na slabší úrovni než zařízení s konfigurací CDC. Zařízení s touto konfigurací disponují virtuálním strojem, který má velikost řádově stovky kilobajtů. Tento virtuální stroj se nazývá KVM (Kilo Virtual Machine).

Typické vlastnosti pro CLDC konfiguraci jsou následující [1]:

- Zařízení je řízeno 16-bitovým či 32-bitovým procesorem s minimální taktovací frekvencí 25 MHz
- Pro běhové prostředí Javy zařízení disponuje minimálně 32 KB RAM a 160 KB ROM (běžně je celková paměť 160KB až 512KB).
- Zařízení může mít omezený zdroj energie (baterie).
- Zařízení je propojitelné s některým typem sítě, pomalejší (typický datový tok 9.6Kbps), přerušitelné spojení.

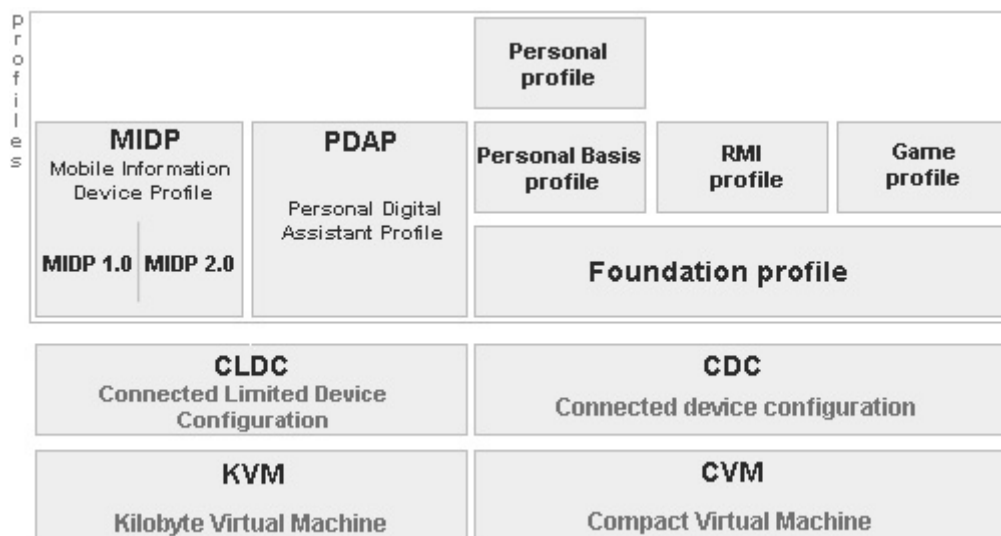
S danými omezeními poskytuje v současnosti CLDC zařízením tyto funkce:

- Soubor základních knihoven Javy (`java.lang`, `java.util`)
- Základní vstup / výstup (`java.io`)
- Základní podporu sítí (`javax.microedition.io`)
- Zabezpečení na úrovni virtuálního stroje, zabezpečení na úrovni aplikace (aplikace mohou volat pouze třídy podporované samotným zařízením – sandbox)

1.3 Profily

Profil je další vrstvou pracovního prostředí J2ME, která doplňuje programovací knihovny využívající příbuzných funkcí v podobných zařízeních. Jedná se o rozlišení různých produktů stejné úrovně (například mobilní telefony disponují jednou konfigurací – CLDC, ale liší se v profilech).

Pod záštitou Java Community Process bylo vytvořeno několik profilů, které jsou specifické nejenom pro produkty různé úrovně, ale i pro určité množiny zařízení na stejné úrovni (pro danou úroveň existuje více profilů). Přehled profilů viz. *obrázek 1.2*.



Obrázek 1.2: Přehled J2ME profilů

Pro programování aplikací pro mobilní telefony se uvažují profily MIDP 1.0 a MIDP 2.0. Zatímco MIDP 1.0 je podporováno jakýmkoli mobilním telefonem, pro MIDP 2.0 to neplatí. Proto je volba profilu při tvorbě aplikace velmi důležitá z důvodu přenositelnosti. V dnešní době však drtivá většina nových mobilních telefonů MIDP 2.0 podporuje [10].

Z důvodu přenositelnosti byla aplikace Sudoku udělána pro mobilní telefony s profilem MIDP 1.0 (využití MIDP 2.0 nebylo nutné).

1.3.1 MIDP 1.0

MIDP 1.0 je základním profilem pro mobilní telefony, který doplňuje programové knihovny pro CLDC konfiguraci. Pokud je žádoucí volná přenositelnost aplikace mezi libovolnými mobilními telefony, je nutné akceptovat tento profil.

MIDP 1.0 definuje následující požadavky na dané zařízení [2]:

- Minimální rozměr obrazovky 96x54 px
- Minimálně 1 bitová barevná hloubka
- Klávesnice pro jednu nebo dvě ruce; nebo doteková obrazovka
- 128 KB stálé paměti pro MIDP komponenty
- 8 KB paměti pro dlouhodobě ukládaná data
- 32 KB nestálé paměti pro běhový chod Javy (JRE, memory heap)
- Obousměrná bezdrátová komunikace

MIDP 1.0 specifikuje následující aplikační rozhraní:

- Správa průběhu aplikací (`javax.microedition.midlet`)
- Uživatelské rozhraní (`javax.microedition.lcdui`)
- Persistentní uchování dat – RMS (`javax.microedition.rms`)
- Obecné síťově připojení (`javax.microedition.io`)

1.3.2 MIDP 2.0

Profil MIDP 2.0 je rozšířením profilu MIDP 1.0 a je s ním zpětně kompatibilní (mobilní telefon s MIDP 2.0 bezproblémově spustí aplikaci napsanou pro MIDP 1.0).

Jelikož obsahuje téměř dvakrát tolik tříd než MIDP 1.0, vzrůstá požadavek na paměť. Jedná se o vzrůst 128 KB RAM (celkem 256 KB) a o 96 KB více pro běh JRE (celkem 128 KB) [2].

Mezi významnější rozšíření programového rozhraní patří:

- Přibyla rozhraní pro další síťové protokoly (např. https, datagram, socket)
- Herní uživatelská rozhraní
- Ovládání zvuku standardním rozhraním
- Sdílení persistentních dat

Jak již bylo řečeno – MIDP 2.0 je podporováno většinou mobilních telefonů. Je však na místě uvažovat i starší mobilní zařízení, která podporují pouze MIDP 1.0 a zda se bez rozšíření MIDP 2.0 lze obejít a tím je zaručena větší přenositelnost aplikace. (Některých vlastností, které jsou přímo pouze v MIDP 2.0, lze nepřímo dosáhnout i v MIDP 1.0. Například v MIDP 2.0 třída *Sprite*, kterážto se používá při tvorbě her, lze naimplementovat na základě technik vyskytujících se v MIDP 1.0.). Aplikace Sudoku je realizována v rámci MIDP 1.0 a je z tohoto pohledu tedy zaručena volná přenositelnost mezi mobilními telefony.

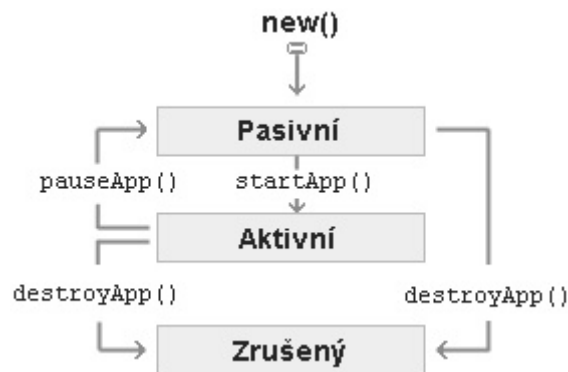
1.4 MIDlet

Při tvorbě mobilní aplikace je nutné zdědit abstraktní třídu `MIDlet`, která se nachází v balíčku `javax.microedition.midlet`. Proto se často mobilní aplikace nazývají midlety.

Při spouštění aplikace je třída, která dědí třídu `MIDlet`, vytvořena zavoláním svého veřejného konstruktoru (V tomto stavu by aplikace neměla používat žádné sdílené zdroje.). Stav, ve kterých se může midlet nacházet, a přechod mezi těmito stavy, ukazuje *obrázek 1.3*. Jedná se o stavy pasivní, aktivní a zrušený. Běh aplikace a přechod mezi stavy řídí **aplikační manažér**.

Při volání metody `startApp()` `MIDlet` inicializuje všechny objekty potřebné pro stav aktivity a nastaví odpovídající obrazovku. Metoda `pauseApp()` se vyznačuje přerušením všech

aktivních vláken. Voláním metody `destroyApp()` by se měly uvolnit nebo zavřít všechny zdroje [9].



Obrázek 1.3: Životní cyklus MIDlet

Příklad ukázky kostry MIDletu:

```
import javax.microedition.midlet.*;

public class MyMIDlet extends MIDlet {
    public MyMIDlet() {
        //konstruktor
    }
    public void startApp() {
        //aktivace midletu
    }
    public void pauseApp() {
        //přerušlení midletu
    }
    public void destroyApp( boolean unconditional ) {
        //zrušení midletu
    }
}
```

2 Algoritmizace Sudoku

Tato kapitola pojednává o obecných vlastnostech Sudoku matic a o algoritmických principech řešení Sudoku a generování nových zadání. Je zde popsáno více variant řešení daného problému a odůvodnění vhodného použití na cílový systém.

2.1 Obecné vlastnosti Sudoku

Sudoku je logická hra, jejíž cílem je doplnit chybějící čísla s určitými omezeními do předvyplněné matice. Herní matice má rozměr $n^2 \times n^2$ (typické Sudoku $3^2 \times 3^2$), kde n značí rozměr submatice (těchto submatic je n^2).

Každé pole výsledné, správně vyplněné, matice musí obsahovat právě jedno číslo, jehož hodnota je v intervalu 1 až n . Přičemž v každém sloupci, řádku a submatici se musí vyskytovat všechna čísla 1 až n . Nemůže tedy nastat, že by se v některém řádku, sloupci či submatici vyskytovala dvě čísla se stejnou hodnotou.

Vyřešená Sudoku matice je speciálním typem *latinských čtverců* (v libovolném řádku a sloupci se vyskytují všechna čísla 1 až n) s doplněním omezením výskytu všech čísel 1 až n v submaticích. Jedná se tedy o podmnožinu latinských čtverců.

Až doposud nebyl objeven obecný vzorec pro vyčíslení počtu možných výsledných Sudoku matic (a latinských čtverců) pro obecný rozměr. Pro dané n se obecně jedná o počet Sudoku matic výrazně menší než počet latinských čtverců. V roce 2005 však *Bertram Felgenhauer* a *Frazer Narcis* vyčíslili počet všech možných vyplněných Sudoku matic pro typický rozměr 9×9 , tento počet je 6 670 903 752 021 072 936 960 [12]. Tedy generátor, který pracuje na principu náhodných (pseudonáhodných) čísel, může poskytnout v každém dalším kroku s vysokou pravděpodobností nové, ještě neřešené zadání.

Z pohledu *grafové teorie* se řešení Sudoku jeví jako problém *obarvení grafu*. Cílem je za pomoci n^2 barev dobarvit graf o $n^2 * n^2$ uzlech (každý uzel reprezentuje místo v Sudoku matici, uzly jsou spojeny s některými jinými uzly hranou – dle Sudoku omezení), jenž má některé uzly již obarveny, tak, aby uzly spojené hranou neměly stejnou barvu. Tato úloha patří mezi *NP-úplné* úlohy.

NP (*nedeterministicky polynomiální*) úloha je množina problémů, které lze řešit v polynomiálně omezeném čase na nedeterministickém turingově stroji - na počítači, který umožňuje v každém kroku rozvést výpočet na n větví, v nichž se posléze řešení hledá současně. *NP-úplné* problémy jsou takové *nedeterministicky polynomiální problémy*, na které jsou polynomiálně redukovatelné všechny ostatní problémy z *NP*. To znamená, že třídu *NP-úplných* úloh (zde spadá Sudoku) tvoří v jistém smyslu ty *nejtěžší* úlohy z *NP*. Neexistuje deterministický polynomiální

algoritmus pro nějakou *NP-úplnou* úlohu (tedy ani pro obecnou Sudoku matici), který by tuto úlohu řešil v polynomiálním čase. [15]

Programová část bakalářské práce je však zaměřena na Sudoku o typických rozměrech 9×9 (nikoliv o obecných rozměrech), u jehož řešení lze uplatnit jistých heuristik – metody, které lidé při řešení používají.

2.2 Řešení Sudoku

Existuje více principů algoritmických řešení Sudoku zadání, přičemž vhodnost použití algoritmu na automatické dořešení Sudoku matice záleží na cílovém systému, na kterém program poběží. Některé algoritmy však nejsou použitelné na řešení Sudoku matic o obecném rozměru ($n^2 \times n^2$).

Tato kapitola bude popisovat různé přístupy algoritmických řešení Sudoku matic.

2.2.1 Řešení hrubou silou

Jedná se o nejjednodušší a nejpomalejší algoritmus řešení Sudoku stylem pokus, omyl. Jeho výhoda spočívá v délce algoritmu a možnosti použít na Sudoku obecných rozměrů (pracuje však pomalu). Princip řešení je následující:

Fáze 1 (určení počátečních kandidátů):

Na začátku (některá čísla jsou již zadána) se určí všichni kandidáti nevyplněných políček. Kandidátem se rozumí číslo, které lze vepsat do příslušné buňky – toto číslo splňuje Sudoku podmínky jednoznačnosti v řádku, sloupci a submatici. V některých políčkách zbude jediný kandidát – toto číslo se stává pevnou hodnotou na dané pozici. Ve zbylých políčkách zůstane dva a více kandidátů. Následuje řešení hrubou silou (pokus, omyl).

Fáze 2 (backtracking):

Vybere se políčko s nejmenším počtem kandidátů (tím se zredukuje šířka prohledávaného stavového prostoru). Z těchto kandidátů se u daného políčka náhodně vybere jeden kandidát a ostatní se zruší (ale pamatují se pro případný zpětný chod). Nově vzniklé jedním číslem ohodnocené políčko ovlivní další políčka a pokračuje se stejným způsobem dále (výpočet nových kandidátů, výběr dalšího kandidáta, ..).

Pokud se algoritmus dostane do stavu, kdy je celá matice vyplněna, skončil úspěšně. Pokud se však algoritmus dostane do stavu, kdy vzniklo políčko (vznikla políčka) s nulovým počtem kandidátů, je nutné se vrátit (*backtracking*) o jeden krok zpět, ustanovit původní stav pro dané políčko a vybrat jiného kandidáta a postupovat opět dále.

Pokud má Sudoku zadání platné řešení, je toto řešení nalezeno. Pokud řešení nemá, dojde při zpětném chodu k vyčerpání všech kandidátů startovacího políčka (a tím i vyčerpání kandidátů všech políček) a algoritmus končí s chybovou indikací řešení nenalezeno.

Jedná se o typický rekurzivní algoritmus typu *Depth First Search*. V praxi se přímo v této podobě příliš nepoužívá, ale je velmi často využíván komplexnějšími algoritmy – viz. následující kapitola 2.2.2 *Řešení s využitím heuristik*.

2.2.2 Řešení s využitím heuristik

Tato metoda řešení patří mezi nejčastěji používané. Její výhoda spočívá v rychlosti, která je zapříčiněna logikou algoritmu, jenž využívá známých heuristik. Heuristikami jsou myšleny principy řešení Sudoku matice za pomoci metod, které lidé při řešení běžně používají. Průběh algoritmu řešení Sudoku matice je následující:

Fáze 1 (Single Candidate):

Dochází k ohodnocení políček, která neobsahují právě jednu hodnotu (jsou prázdná, nebo obsahují dva či více kandidátů), kandidáty dle Sudoku podmínek. Pokud v některém políčku zůstane jediný kandidát, stává se právoplatným číslem na dané pozici a *fáze 1* se opakuje (Políčko, kde nově vzniklo jediné číslo, může ovlivnit další kandidáty.). Pokud bude *fáze 1* neúspěšná (nevznikl nový jediný kandidát), přechází se na *fázi 2*.

Fáze 2 (Single in Row/Column/Box):

V této fázi je mapováno, zda se některá hodnota z kandidátů vyskytuje v řádku, sloupci nebo submatici právě jednou – dochází k porovnání s pevně vyplněnými čísly i kandidáty.

Například na *obrázku 2.1* políčko na řádku 9 a sloupci 7 (dále se bude značit R9C7, ve smyslu RřádekCsloupec) obsahuje kandidáty 1,2,6. Pokud se tyto kandidáty porovnají s jinými kandidáty ve sloupci, řádku nebo submatici, zjistí se, že hodnota 6 je jedinou hodnotou zde se vyskytující (stačí, aby byla jednoznačná pouze ve sloupci, pouze na řádku nebo pouze v submatici). Stane se tedy jediným číslem, které může na pozici R9C7 být (kandidáti 1 a 2 se odstraní).

4		5				8		
					7	123		
8	7	9			4	123	6	
5	8	4	2	7	1	9	3	6
	2		9			1		4
				4		12	5	
9	4	2	7	1	6	5	8	3
6	1	8	3	2	5	4	9	7
7	5	3	4			12	12	12

Obrázek 2.1: Jediný kandidát ve sloupci

Pokud je tato 2. fáze úspěšná, je znovu aplikována *fáze 1*, jinak se pokračuje na *fázi 3*.

Fáze 3 až N-1:

Úspěchem předcházejících dvou fází bylo vždy výsledné jediné číslo na určité pozici. Nyní následují fáze, jejichž výsledkem obecně nemusí být jediné číslo, ale dochází k redukci kandidátů.

Pravidel je velké množství [13] a záleží na účelu aplikace, která se zvolí. Některá pravidla jsou zřídka úspěšně použitelná a jejich vykonáváním se spíše ztrácí čas. Velkým množstvím pravidel se také zvětšuje algoritmus a tím i velikost výsledného programu, což se jeví jako důležitý aspekt pro mobilní telefony. Je tedy vhodné učinit vhodný kompromis při výběru pravidel s ohledem na cílové zařízení.

Mezi relativně úspěšná pravidla patří například *Naked Pairs*. Principem tohoto pravidla je nalezení dvou políček ve sloupci, v řadě a nebo v submatici, kde se vyskytují v každém ze dvou políček dva kandidáti o stejných hodnotách. Například na *obrázku 2.2* se na pozicích R5C7 a R6C7 (zvýrazněny zeleně) vyskytují právě dva kandidáti o stejných hodnotách 6, 9. Je zřejmé, že jedno z políček bude mít výslednou hodnotu 6, druhé pak 9. Z toho plyne, že ve zbývajících políčkách v příslušném sloupci a příslušné submatici musí být hodnoty 6 a 9 odstraněny (budou odstraněny z políček R2C7, R3C7, R6C8, R8C7, R9C7).

7		2		9		3		
8		5				¹ 4	⁴ 5	
1		9				⁴ 5	⁸ 9	
	9			6		7	4	² 3
						¹ 6	⁵ 8	
						¹ 6	² 3	⁹ 1
	7					2		
3			5			¹ 4	⁸ 9	
			1			⁴ 5	⁸ 9	

Obrázek 2.2: *Naked Pairs*

Pokud dojde k úspěšné aplikaci *Fáze 3 až N-1*, chod algoritmu se opět vrací k *fázi 1*. Pokud je aktuálně prováděná fáze neúspěšná, pokračuje se fází následující.

Fáze N (Nishio):

Úloha řešení Sudoku matice není algoritmicky kompletně pokryta na základě předchozích heuristik – mohou zůstat někteří kandidáti, kteří nebyli odhaleni na základě předešlých metod. Je tedy nutné použít metodu, která dokáže dořešit libovolný správný stav Sudoku. Používá se metoda, která matici dořeší hrubou silou – tedy ta metoda, která byla popsána v kapitole 2.2.1 *Řešení hrubou silou* – *fáze 2*. Této metodě se říká *Nishio*.

2.2.3 Knuthův algoritmus X

Na řešení problémů typu *zaplnění systému útvarů* (*exact cover*, dále v některých případech označované *EC*) se používá *Knuthův algoritmus X*. Jedná se o rekurzivní, nedeterministický, prohledávající do hloubky (*depth-first search*) algoritmus využívající hrubou sílu (*brute-force*) [17].

Pro *zaplnění systému útvarů* platí [16]:

Nechť U je univerzum elementů, S kolekce množin. *Zaplnění systému útvarů* je podkolekce S^* kolekce S taková, že každý prvek universa U je obsažen v právě jedné množině kolekce S^* .

Sudoku se svými omezujícími podmínkami mezi problémy typu *EC* patří. Z pohledu *EC* je účelné uvažovat následující Sudoku omezení:

- Každé políčko matice musí obsahovat právě jedno číslo.
- Každý řádek musí obsahovat každé číslo právě jednou.
- Každý sloupec musí obsahovat každé číslo právě jednou.
- Každý submatice musí obsahovat každé číslo právě jednou.

Všechna omezení se zaznačí jako matice s hodnotami 0 a 1, viz následující obrázek (pro typické Sudoku 9×9)

	právě jedno číslo v políčku			omezení pro řádek			omezení pro sloupec			omezení pro submatici		
	R1C1	R1C2	...	R1#1	R1#2	...	C1#1	C1#2	...	B1#1	B1#2	...
R1C1#1	1	0	...	1	0	...	1	0	...	1	0	...
R1C1#2	1	0	...	0	1	...	0	1	...	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...	...
R9C9#9	0	0	...	0	0	...	0	0	...	0	0	...

Obrázek 2.3: matice zaplnění systému útvarů

První sloupec na obrázku 2.3 značí všechny možné kandidáty. Například *R1C1#2* značí, že v Sudoku matici na prvním řádku a prvním sloupci je hodnota dvě. Jelikož má Sudoku matice 9×9 políček a na každém políčku může být 9 kandidátů, je celkový počet kandidátů (a tím i řádků *EC* matice) $9 \times 9 \times 9 = 729$.

První řádek na obrázku 2.3 značí různá omezení v Sudoku matici. Například *R1C1* značí ohodnocení políčka, které je na prvním řádku a prvním sloupci (rozhodnutí, které číslo na této pozici bude). *R1#1* značí ohodnocení prvního řádku Sudoku matice číslem 1 (rozhodnutí, ve kterém sloupci na prvním řádku číslo 1 bude). *C1#1* značí ohodnocení prvního sloupce Sudoku matice číslem 1. *B1#1* značí ohodnocení první submatice Sudoku matice číslem 1. Těchto omezujících množin je dohromady $9 \times 9 + 9 \times 9 + 9 \times 9 + 9 \times 9 = 324$.

Výsledná *EC* matice má tedy rozměr 729×324 .

EC matice je ohodnocena čísly 0 a 1. 1 značí, že se daný kandidát na dané pozici vyskytuje. Úspěšným dořešením Sudoku matice je dostat EC matici do stavu, kdy se v každém sloupci vyskytuje právě jedna hodnota 1 (vybírání se podmnožina řádků). Tento problém řeší Knuthův algoritmus X.

Knuthův algoritmus X [17]:

Nechť EC problém (řešení Sudoku) je reprezentován maticí A , která obsahuje hodnoty 0 a 1.

- 1) Jestliže matice A je prázdná, problém je vyřešen. Algoritmus končí úspěchem.
- 2) V jiném případě vyber sloupec c , deterministicky.
- 3) Vyber řádek r tak, že $A_{r,c} = 1$.
- 4) Uvažuj řádek r jako částečné řešení.
- 5) Pro každý sloupec j tak, že $A_{r,j} = 1$,
smaž sloupec j z matice A ;
pro každý řádek i takový, že $A_{i,j} = 1$,
smaž řádek i z matice A .
- 6) Opakuj tento algoritmus rekurzivně – redukce matice A .

V druhém kroku se vybírá takový sloupec, který obsahuje nejmenší počet 1 (redukce prohledávacího stavového prostoru).

Implementace algoritmu X se velmi často realizuje pomocí techniky *tančících propojů* – *Dancing Links (DLX)*.

Algoritmus X se většinou používá na problémy s větším počtem omezujících podmínek. Pro řešení Sudoku matice o rozměru 9×9 dává podobné časové výsledky jako „obyčejný“ backtracking popsaný v kapitole 2.2.1 *Řešení hrubou silou*. Jeho výhoda pro aplikaci na řešení Sudoku tkví v použitelnosti na větší rozměry Sudoku matice (zde dává výrazně rychlejší časy než „obyčejný“ backtracking).

2.3 Generování Sudoku zadání

Tato kapitola bude popisovat principy generování Sudoku zadání. Výrazná většina generátorů je velmi silně závislá na algoritmech, jejichž účelem je Sudoku matice vyřešit (viz. předchozí kapitola 2.2 *Řešení Sudoku*).

2.3.1 Využití existujících Sudoku zadání

Jedná se o nejjednodušší algoritmus na generování Sudoku zadání. Vychází se z již existujícího zadání. V dané matici dochází k výměně řádků a sloupců. Tato výměna však probíhá tak, aby byla zachována všechna omezení pro čísla. Nesmí dojít k přesunu čísel do jiné submatice. Mohou se tedy

mezi sebou měnit řádky či sloupce 1 až 3 nebo 4 až 6 a nebo 7 až 9 (nelze tedy vyměnit například 3. a 4. řádek).

Tento algoritmus je rychlý, ale produkuje omezený počet matic a matice jsou nerozlišené obtížnosti.

2.3.2 Generování s využitím algoritmu pro řešení Sudoku

Tato technika generování je silně závislá na principu řešení Sudoku. Je tedy značně důležité udělat efektivní algoritmus pro dořešení Sudoku matice (pro označení *algoritmu pro dořešení Sudoku zadání* bude dále používáno *solver*). Jak již bylo zmíněno, jedná se o nejčastěji používaný princip generování. Navíc je to jediný způsob, jak vygenerovat zadání o příslušné obtížnosti. Existují dva základní přístupy.

2.3.2.1 Postupné odkrývání čísel

Náhodně se vygeneruje N čísel tak, aby tato čísla splňovala všechny omezující Sudoku podmínky. Pro číslo N platí, že je jeho hodnota minimálně 17. Důvod je ten, že neexistuje Sudoku, které by mělo méně než 17 čísel. Tohle není však matematicky doloženo (nebyl nalezen žádný důkaz) a jedná se pouze o experimentální tvrzení.

Nyní se náhodně vygenerovaná matice dořeší – použije se solver (řešení je obecně více, stačí nalézt jedno). Pokud není řešení nalezeno, generuje se N nových čísel, jinak se pokračuje.

V tomto stavu je matice celá vyplněna (nese hodnoty „budoucího“ řešení). Nyní dochází k postupnému odstraňování čísel na náhodných pozicích (Pokud má mít výsledné zadání symetrické rozmístění čísel, pak se odstraňují čísla na místech k sobě symetrických.). V každém kroku se však aplikuje solver s metodou backtracking (nebo Algoritmus X), aby se zjistilo, zda má matice v aktuálním stavu právě jedno řešení (jelikož korektní Sudoku zadání má mít právě jedno řešení). Pokud má matice v příslušném stavu více řešení, pak se odstraněné číslo vrací zpět a buď se vybírá číslo jiné, a nebo pokud je matice v požadovaném stavu, algoritmus úspěšně končí. Požadovaným stavem je myšleno, že se aplikuje solver a zjistí se, které řešící metody byly použity – z toho je odvozena obtížnost vygenerovaného Sudoku zadání (Pokud šla matice v posledním kroku dořešit za pomoci jednodušších metod, jedná se o lehké zadání, apod.).

2.3.2.2 Postupné přidávání nových čísel

Vygeneruje se N náhodných čísel (opět $N \geq 17$), tato čísla splňují Sudoku podmínky.

Nyní se matice dořeší za pomoci solver. Pokud řešení není nalezeno, generuje se N nových čísel. Pokud je nalezeno právě jedno řešení, končí se úspěšně. Jestliže je nalezeno více řešení, vezmou se dvě různá řešení. Tato dvě řešení se na některých pozicích liší v hodnotách. Vybere se náhodně některé místo, kde se liší, a k počátečním N číslům se přidá číslo další, které se nachází na vybraném

místě (není důležité, ze kterého ze dvou řešení se použije) – nyní je $N+1$ čísel. Opět se aplikuje solver na počátečních $N+1$ čísel.

Obtížnost vygenerované matice se odvíjí od náročnosti aplikací metod na závěrečné dořešení matice.

3 Midlet Sudoku

V rámci bakalářské práce byla vytvořena hra Sudoku pro mobilní telefony. Tato kapitola bude popisovat jednotlivé aspekty a principy tvorby aplikace. Nejedná se o popis na bázi uživatelské příručky, jak zacházet s aplikací – tento popis se nachází v příloze 1 *Manuál použití aplikace*. Jedná se o implementační popis. Jak byly některé vlastnosti aplikace realizovány, zdůvodnění realizace a jak by se daný problém mohl eventuálně řešit jinak.

3.1 Předběžné aspekty před tvorbou aplikace

Programátor, který vyvíjí aplikaci pro mobilní telefony, si musí v první řadě uvědomit, pro jaké cílové uživatele tato aplikace bude, respektive pro jaké typy mobilních telefonů. Již v kapitole 1.1 *Obecný pohled na J2ME* bylo zmíněno, že ačkoliv se obecně Java ve svých edicích snaží o jednotnou specifikaci, v edici J2ME se tomu z důvodů technických odlišností zařízení musí vyhnout. Specifikace je dána až určením dvojice konfigurace a profilu.

Jelikož je bakalářská práce zaměřena pro mobilní telefony, je výběr konfigurace snadný. Konfigurace určená pro mobilní telefony je v drtivé většině CLDC. Existuje však technické rozlišení na úrovni CLDC konfigurace, a to CLDC verze 1.0 a verze 1.1. Hlavní rozdíl pro programátora spočívá v tom, že verze 1.1 používá plovoucí řádovou čárku. Aplikace Sudoku však pracuje pouze s celými čísly, je tedy na místě realizace v souladu s konfigurací CLDC 1.0.

Důležitější rozhodnutí je na úrovni profilu. Pro mobilní telefony se uvažují profily MIDP 1.0 a MIDP 2.0 (někdy se u mobilních aplikací používá označení Java 1, Java 2). Jelikož je využití vlastností, které nabízí MIDP 2.0 (viz kapitola 1.3.2 *MIDP 2.0*), v aplikaci Sudoku zbytečné, je výběr profilu MIDP 1.0 vhodnější. Tím se zaručí i větší přenositelnost mezi různými mobilními telefony. Pokud se však při tvorbě aplikace jeví účelné využití některých vlastností MIDP 2.0 a není-li nutné aplikaci přenášet (portovat) na starší mobilní telefony (všechny nové mobilní telefony totiž pracují s MIDP 2.0), je vhodné toho využít.

Dalším důležitým aspektem je rozlišení displeje. Pokud má být aplikace spustitelná na větším počtu mobilních telefonů s různými rozlišeními displeje, jsou zde dva základní principy tvorby aplikace, které se v praxi uplatňují. Jeden princip je založen na tom, že se vytvoří více variant jedné aplikace – tyto varianty se budou lišit pouze v rozlišení zobrazení (aplikace obsahuje pouze jednu sadu obrázků o pevném rozlišení). Pokud si uživatel tuto mobilní aplikaci stahuje, musí uvést typ mobilního telefonu (a tím je určena i velikost rozlišení), pro který je určena. Tyto aplikace jsou obecně graficky příjemnější (jsou šity na míru) a zabírají méně místa. Druhá varianta je pracnější a spočívá v tom, že se udělá jedna jediná aplikace, která se adaptuje na cílové rozlišení. Toho se docílí tak, že se použije více sad stejných obrázků, které se liší pouze v rozměrech (konkrétní obrázek se

použije až za běhu, kdy aplikace zná rozlišení cílového zařízení), nebo se použije vestavěné vysokoúrovňové grafiky (formuláře, textová pole, ..) a grafických geometrických primitiv z nízkoúrovňové grafiky (obdélníky, úsečky, ..), kterým se udávají rozměry až za běhu (a nebo kombinace dvou předešlých technik). Obecná nevýhoda druhého řešení tkví v tom, že se pak jedná o graficky méně příjemnou aplikaci (použití grafických geometrických primitiv), nebo má aplikace větší velikost (použití více sad obrázků). Výhodou však je, že se jediná aplikace graficky přizpůsobí cílovému zařízení. Pro získání zkušeností s adaptovatelností aplikace na cílové rozlišení displeje, je aplikace Sudoku ubírána touto cestou – využití obrázku a vestavěné vysokoúrovňové / nízkoúrovňové grafiky (popis konkrétního docílení této techniky více v kapitole 3.2.1.2 *Uživatelské rozhraní*).

Pokud je mobilní aplikace určená pro CLDC 1.0, MIDP 1.0 a obecné rozlišení displeje, jedná se o takzvanou **generickou** aplikaci (z pohledu konfigurace, profilu a zobrazení je aplikace přenositelná na kterýkoliv typ mobilního telefonu). Tato omezení jsou nejobecnějšího charakteru. Je nutné však uvažovat ještě další vlastnosti, které budou probírány v dalších kapitolách.

Při vývoji aplikace není na škodu se inspirovat vlastnostmi cizích (komerčních) aplikací. Aby byla aplikace na trhu úspěšná, musí přirozeně nabízet „něco navíc“ oproti existujícím aplikacím. Aplikace Sudoku tímto disponuje – viz. kapitola 3.2.4 *Import zadání z webového serveru*.

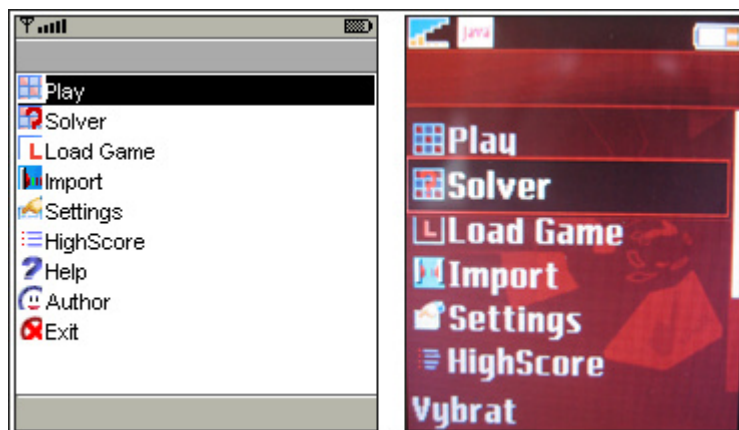
3.2 Spuštění aplikace a úvodní menu

Při spuštění midletu Sudoku je zavolána hlavní spouštěcí třída, která dědí abstraktní třídu MIDlet. Což je povinností jakékoliv mobilní aplikace (viz kapitola 1.4 *MIDlet*). Tato třída implementuje povinné metody `startApp()`, `pauseApp()` a `destroyApp()`.

Vykonáním metody `startApp()` dochází k inicializaci displeje a zobrazením úvodního menu. Menu obsahuje následující položky (popis implementace v následujících kapitolách):

- **Play** (generace nové hry Sudoku)
- **Solver** (automatický řešitel Sudoku zadání)
- **Load Game** (dohrání rozehrané uložené hry)
- **Import** (Import zadání z webového serveru)
- **Settings** (nastavení vlastností hry)
- **HighScore** (časová statistika úspěšnosti hráčů)
- **Help** (pravidla řešení hry Sudoku a vysvětlení ovládání hry)
- **Author** (informace o autorovi aplikace)
- **Exit** (ukončení aplikace)

Grafická reprezentace menu v emulátoru J2ME Wireless Toolkit a mobilním telefonu Sony Ericsson K750i je na následujícím *obrázku 3.1*.



Obrázek 3.1: Reprezentace menu v emulátoru J2ME Wireless Toolkit a mobilním telefonu Sony Ericsson K750i

Menu se v mobilních aplikacích obecně řeší dvěma způsoby. Buď nízkouúrovňovou grafikou (obrázky), nebo vysokoúrovňovou grafikou (seznam položek). V Sudoku aplikaci je menu vyřešeno vysokoúrovňovou grafikou, což zaručuje vyšší přenositelnost na mobilní telefony s různými rozlišeními displeje a jednodušší udržovatelnost (lze velmi lehce dosáhnout rozšíření / modifikace menu). Jedná se o seznam položek typu `javax.microedition.lcdui.List`.

Jednotlivé položky jsou ve formátu ikonka (obrázek) a popisek. Mobilní telefony se dosti značně liší v maximální velikosti obrázkové ikonky, což se jeví jako určité omezení. V aplikaci Sudoku jsou použity ikonky o velikosti 16×16 px a v některých zařízeních jsou tyto ikonky ořezány (viz. emulátor na obrázku 3.1). Programátorsky se nedá zjistit velikost ikony podporované na cílovém zařízení. Nelze tedy za běhu aplikace zjistit podporovanou velikost ikony cílovým zařízením, mít vytvořenou sadu stejných ikon, které se liší pouze rozlišením, a v průběhu spuštění aplikace vybrat příslušnou sadu o požadovaném rozlišení.

Pokud uživatel vybere nějakou položku z menu (seznamu), zavolá aplikační manažér příslušného posluchače událostí, kde se zjistí, která položka byla vybrána.

3.2.1 Generování nové hry Sudoku

3.2.1.1 Generování zadání

Generování Sudoku zadání probíhá ve třech stupních obtížnosti (dle zvolení uživatelem v **nastavení** aplikace - **settings**) – lehká (*easy*), střední (*intermediate*), těžká (*hard*). Přičemž lehké zadání je takové, které lze vyřešit na základě metody *Single Candidate*. Střední zadání lze vyřešit použitím metod *Single Candidate* a *Single in Row/Column/Box*. Na vyřešení těžkého zadání je nutné použít i metod těžšího charakteru (uživatel si nevystačí s metodami *Single Candidate* a *Single in Row/Column/Box*). Tyto metody jsou popsány v kapitole 2.2.2 Řešení s využitím heuristik.

Byly implementovány a otestovány oba algoritmy popsané v kapitole 2.3.2 *Generování s využitím algoritmu pro řešení Sudoku*, přičemž vestavěný solver byl implementován dle kapitoly 2.2.2 *Řešení s využitím heuristik*.

Jelikož se v poslední fázi aplikace *solveru* využívá backtracking (v momentě, kdy jsou všechny heuristiky vyčerpány), dochází k rekurzivnímu prohledávání a ohodnocování stavového prostoru. Aby však nedocházelo k hlubokému zanoření, po určitém N-tém kroku (bylo experimentem stanoveno číslo 250) rekurze dochází k návratu a automatickému generování nového zadání. Bylo experimentem ověřeno, že se jedná o rychlejší variantu a nedochází k občasným hlubokým zanořování rekurze.

Testování rychlosti generování bylo prováděno na mobilním telefonu Sony Ericsson K750i, který disponuje procesorem ARM9 s taktovací frekvencí 112 MHz. Dále pro porovnání bylo taktéž testováno na osobním počítači s procesorem Pentium M s taktovací frekvencí 1.6GHz (FSB 400MHz, L2 2MB). Testovací aplikace generovala v cyklu 20 zadání požadované obtížnosti. Aplikace byla spuštěna pro každou obtížnost třikrát.

V prvním případě byl naimplementován a testován algoritmus generování nového zadání, který je popsán v kapitole 2.3.2.2 *Postupné přidávání nových čísel*. Doba generování Sudoku zadání lehké a střední obtížnosti byla relativně přijatelná. Generování těžké obtížnosti již bylo nepřijatelné. Naměřené časy jsou na následujícím obrázku 3.2.

zařízení obtížnost	SE k750i - 112MHz		Intel Pentium M 1.66GHz	
	20 zadání	Ø 1 zadání	20 zadání	Ø 1 zadání
lehká (easy)	31 545 ms	1 577 ms	938 ms	47 ms
	39 473 ms	1 974 ms	969 ms	49 ms
	35 995 ms	1 800 ms	766 ms	38 ms
střední (intermediate)	41 479 ms	2 074 ms	1 515 ms	76 ms
	60 413 ms	3 021 ms	1 453 ms	73 ms
	47 227 ms	2 361 ms	1 547 ms	77 ms
těžká (hard)	319 826 ms	15 991 ms	8 031 ms	402 ms
	216 805 ms	10 840 ms	5 922 ms	296 ms
	418 438 ms	20 922 ms	7 360 ms	368 ms

Obrázek 3.2: Doba generování za pomoci algoritmu popsaného v kapitole 2.3.2.2 *Postupné přidávání nových čísel*.

Ve druhém případě byl implementován algoritmus dle kapitoly 2.3.2.1 *Postupné odkrývání čísel*. Časy pro všechny obtížnosti jsou již obstojné, viz obrázek 3.3. Tato implementace se jeví rychlejší a je proto použita v mobilní aplikaci Sudoku.

Jestliže některá strana displeje má menší rozměr než 170px, je matice vykreslena pomocí nízkoúrovňové grafiky – pomocí základních geometrických primitiv (úsečky, obdélníky). Rozměry geometrických primitiv se vypočítají při startu aplikace a jsou přirozeně odvozeny z velikosti displeje.

Pro zobrazení čísel není možné použít vestavěných fontů, jelikož nelze určit přesná velikost tohoto fontu. V J2ME lze využít pouze fontů s relativními rozměry – SIZE_SMALL, SIZE_MEDIUM, SIZE_LARGE; přičemž pouze platí SIZE_SMALL < SIZE_MEDIUM < SIZE_LARGE (ale nelze určit velikost fontu). Konkrétní velikost vestavěných fontů je závislá na zařízení, na kterém aplikace běží. Z tohoto důvodu byly udělány tři sady čísel ve formě obrázků – tyto obrázkové sady se liší v rozměru. Příslušný rozměr se použije za běhu aplikace. Vykreslování textu pomocí obrázků je dokonce rychlejší než vykreslování textu pomocí vestavěných fontů. Zobrazení herní matice pro displej s rozlišením 130 x 130px je znázorněno na obrázku 3.5.



Obrázek 3.5: Zobrazení herní matice na displeji s rozlišením 130 x 130 px využitím nízkoúrovňové grafiky – pomocí geometrických primitiv a čísel ve formě obrázků.

Jelikož se v obou případech pro zobrazení používá nízkoúrovňové grafiky (obrázky, grafická geometrická primitiva), jsou vstupy od uživatele (pohyb po matici, zadávání čísel, vyvolání herního menu) reprezentovány jako nízkoúrovňové události.

K indikaci stisku konkrétní klávesy slouží metody pro oznámení události (stisk klávesy, uvolnění klávesy, dvojitý stisk klávesy). Každá klávesa je mapována na konkrétní klávesový kód – celé číslo. Existuje určitá množina kláves, které musí být obsaženy na libovolném mobilním telefonu (numerické klávesy, #, *, ..). A právě k těmto klávesám existuje ekvivalentní vyjádření pomocí J2ME klávesových konstant. Z důvodu přenositelnosti aplikace Sudoku je těchto vestavěných konstant použito. Je nutné však řešit problém pohybu po matici – některé mobilní telefony pro pohyb do stran používají joystick, speciální klávesy, apod. Různé docílení pohybu na různých mobilních telefonech obecně mapuje jiné číselné konstanty. Ve specifikaci J2ME jsou však definovány herní akce pomocí jednoznačných herních číselných konstant. Zatímco každý klávesový kód je mapován pro jednu herní akci, herní akce může být spojena s více než jedním klávesovým kódem.

3.2.2 Automatický řešitel Sudoku zadání

Automatický řešitel Sudoku zadání slouží k dořešení Sudoku matice, která je zadána uživatelem. Algoritmus, který tento problém řeší, byl implementován dle kapitoly 2.2.2 *Řešení s využitím heuristik*. Jako heuristické metody byly použity *Single Candidate*, *Single in Row/Column/Box*, *Naked Pairs* (tyto metody jsou popsány v kapitole 2.2.2 *Řešení s využitím heuristik*) a *Hidden Pairs*.

Hidden Pairs je metoda spočívající v nalezení dvou políček – v řádku, sloupci, či submatici, která obsahují dva shodné kandidáty. Tato políčka můžou narozdíl od metody *Naked Pairs* obsahovat i jiné kandidáty. Pokud se ve zbývajících políčkách v řadě, sloupci, či submatici nevyskytují hodnoty, které jsou rovny dvěma shodným kandidátům ve vybraných políčkách, pak dojde ke zrušení zbývajících kandidátů v těchto dvou políčkách. Aplikaci *Hidden Pairs* demonstruje obrázek 3.6. R3C6 a R3C9 obsahují shodné kandidáty 1 a 3. Jelikož se tyto hodnoty v řádku již nevyskytují, je zřejmé, že políčka R3C6 a R3C9 musí hodnoty 1 a 3 obsahovat (není nyní důležité, které políčko bude obsahovat kterou konkrétní hodnotu). Dojde tedy k vyloučení zbývajících kandidátů z těchto dvou políček – v tomto případě kandidáta 2 z políčka R3C9.

8		1			6		9	4
3					9		8	
9	7	4	6	4	2	8	1	3
5	4	7			6	2		3
6	3	2					5	
1	9	8	3	7	5	2	4	6
	8	3	6	2		9	1	5
	6	5	1	9	8			
2	1	9	5					8

Obrázek 3.6: Metoda řešení Sudoku matice - *Hidden Pairs*

Pokud má uživatelem zadaná matice řešení, je toto řešení vždy nalezeno a ukázáno. Jestliže má zadaná matice řešení více (správné zadání má být řešitelné pouze jedním postupem – nejedná se tedy o korektní Sudoku zadání), je zobrazeno první nalezené řešení. Pokud řešení nemá, je indikováno, že matice nelze vyřešit.

Naimplementovaný solver byl opět testován na mobilním telefonu Sony Ericsson K750i s procesorem 112MHz a na osobním počítači s procesorem Intel Pentium M 1.6GHz. Následující obrázek 3.7 ukazuje porovnání časů na těchto dvou různých zařízeních při různých stupních obtížnosti zadání. Hodnoty časů jsou průměrné. Na mobilní telefonu byl solver spuštěn 10x, na osobním počítači 1000x.

sítě), avšak z důvodu technické realizace sítě jsou nižší vrstvy v plné kompetenci mobilní sítě operátora (zde tedy není jednoznačný přístup) [9].

Z důvodu generičnosti aplikace Sudoku byla komunikace implementována pomocí protokolu HTTP. Je však nutné mít na mobilním telefonu nastavený přístupový bod (Access Point) na Internet, nikoliv na WAP. Pokud by byl nastaven přístupový bod na WAP, muselo by mít zařízení k dispozici bránu, která by byla mezi zařízením s WAP a serverem. Tato brána by sloužila pro DNS rozpoznávání URL jmen.

Při požadavku uživatele importování nového zadání, se naváže HTTP komunikace s webovým serverem, konkrétně se stránkou <http://www.tsholesov.cz/sudoku/getSudoku.php>. Zde probíhá generace nového zadání dle kapitoly 4 *Webová aplikace* (pokud vznikne v daném týdnu/roce požadavek importování zadání, je v tomto týdnu/roce vygenerováno právě jedno zadání). Mobilní telefon proudem HTTP přijímá Sudoku zadání, označení týdne a roku, pro který je Sudoku zadání určené. Sudoku zadání je v daném týdnu/roce vždy stejné – jediné.

Jakmile uživatel Sudoku matici vyřeší, posílá se zpět na webový server jméno řešitele, čas řešení, týden a rok, pro který byla matice určena. Musí se zpět odesílat týden a rok, pro které je zadání určené (i když se to dá spočítat z aktuálního data), jelikož je možné, že si uživatel nechá poslat zadání před koncem týdne, ale matici vyřeší až po době, kdy týden skončil (došlo by tedy k uchování skóre pro špatný týden/rok). Odesílající řetězec je ve formátu http://www.tsholesov.cz/sudoku/insertScore.php?week=ČÍSLO_TÝDNE&year=ČÍSLO_ROKU&scoreTime=DOBA_ŘEŠENÍ&nick=JMÉNO_ŘEŠITELE. Tedy komunikace s webovým serverem probíhá pomocí GET hlaviček. Tento odesílací řetězec je interní záležitostí aplikace a není veřejný. Toho by se v opačném případě dalo zneužít fiktivním odesíláním dat na webový server. K silnějšímu potlačení tohoto nešvaru by bylo vhodné na odesílající data použít šifrovací funkci (webový server by pak aplikoval dešifrovací funkci).

Na webovém serveru se výsledky řešení kumulují a zobrazují ve formě tabulek, kde se uživatel může porovnat s ostatními.

3.2.5 Seznam časů úspěšných řešitelů

Ukládání výsledků je realizováno pomocí RMS.

Záznamy s výsledky je nutné seřadit od nejlepšího skóre k nejhoršímu skóre. Řadící algoritmus databázových záznamů je interní záležitostí J2ME. Programátor pouze implementuje porovnávací relaci dvou záznamů (menší, větší, rovná se), který je vestavěným řadícím algoritmem využit. Pokud je řazeno větší množství záznamů, dochází k nemalé časové prodlevě. Z tohoto důvodu je v aplikaci Sudoku seznam řešitelů omezen na 20 položek s nejlepším časovým ohodnocením.

3.2.6 Nastavení, Náповěda, Autor

Položky nastavení, nápověda a autor byly implementovány pomocí vestavěné vysokoúrovňové grafiky (formuláře, textová pole, ..). Nedochází tedy k žádnému omezení při spuštění na libovolném mobilním telefonu, který podporuje J2ME.

4 Webová aplikace

Kapitola zdůvodňuje účel tvorby webové aplikace a způsob ukládání herních výsledků a generování nových Sudoku zadání.

4.1 Účel webové aplikace

Ukládání výsledků v mobilním telefonu nese omezení v možnosti porovnat se s ostatními uživateli. Z tohoto důvodu byla implementována webová aplikace, která slouží k ukládání výsledků řešitelů Sudoku matic. Webová aplikace byla programována technologiemi PHP a MySQL.

Pro objektivnější porovnání časů uživatelů je pro daný týden / rok vygenerované jediné zadání. Tedy pokud si libovolní dva uživatelé ve stejném týdnu / roce nechají importovat zadání, dostanou stejnou Sudoku matici. Tím jsou zaručeny stejné podmínky pro všechny řešitele.

Jakmile uživatel importovanou Sudoku matici vyřeší, pošle se zpět na webový server jméno řešitele a čas řešení. Zde dochází k ukládání výsledků a souhrnnému zobrazení, kde se uživatel může porovnat s ostatními hráči.

4.2 Způsob generování nových zadání

Jakmile uživatel zadá požadavek importovat zadání z webového serveru, dochází k ověření, zda pro daný týden / rok zadání již existuje. Pokud zadání neexistuje (tento uživatel je prvním žadatelem v daném týdnu / roce) dochází ke generování nového zadání, uložení do MySQL databáze pro případné další uživatele a odeslání žadateli k řešení. Pokud zadání existuje, nedochází ke generaci nového zadání, ale použije se zadání, které je pro daný týden / rok uloženo v MySQL databázi (bylo vygenerováno prvním uživatelem v daném týdnu / roce).

Generované zadání má náhodnou obtížnost. Generátor napsaný v PHP pracuje podle algoritmu popsaného v kapitole 2.3.2.2 *Postupné přidávání nových čísel* (pro počítačové platformy dává výrazně lepší časové výsledky než pro mobilní telefony se slabšími procesory). Při nespecifikované požadované obtížnosti generuje zadání s rovnoměrnějším rozložením obtížnosti (nepřevažuje výrazně jediná obtížnost) než generátor implementovatelný podle kapitoly 2.3.2.1 *Postupné odkrývání čísel*.

Po vyřešení importované Sudoku matice se jméno řešitele a čas řešení pošle zpět na webový server. Pokud se jedná o prvního úspěšného řešitele v týdnu / roce, dochází k nové tvorbě MySQL tabulky, která bude uchovávat hodnoty úspěšných řešitelů pro daný týden / rok, a uložení času a jména řešitele do této tabulky. Pokud pro daný týden / rok již úspěšný řešitel existuje, dochází pouze k uložení nových dat do MySQL databáze (tabulka je již vytvořena).

5 Porovnání s cizími aplikacemi

Mobilní aplikace Sudoku, která byla vytvořena v rámci bakalářské práce (dále označované SudokuBP), byla porovnána s cizími mobilními aplikacemi Sudoku. Bylo porovnáno s 9 aplikacemi, přičemž 4 z nich byly komerční (dále označované SudokuKOM), 5 nekomerční (dále označované SudokuNEKOM).

Menu SudokuKOM je graficky propracovanější než SudokuBP – menu je kompletně vytvořeno pomocí obrázků. Testovaná SudokuKOM jsou však určena pro konkrétní typ mobilního telefonu. SudokuBP bylo vytvořeno v rámci maximální generičnosti (nemusí se určovat konkrétní typ mobilního telefonu) a pokud by se použilo kompletně obrázkové menu, musela by aplikace obsahovat více sad stejných obrázků lišící se pouze velikostí, čímž by vzrostla velikost aplikace. Menu SudokuNEKOM bylo na podobné bázi jako SudokuBP.

Pokud je SudokuBP spuštěno na mobilním telefonu s rozlišením minimálně 170 x 170 px, je herní matice zobrazena pomocí obrázku, v jiném případě je využito grafických geometrických primitiv – dojde k přizpůsobení velikosti displeje. Zobrazení pomocí obrázků je pro SudokuKOM typické. Některé SudokuNEKOM využívaly obrázky, některé geometrických primitiv. Pokud byly u SudokuNEKOM užity obrázky, docházelo často na menších displejích k ořezání (nebyl zobrazen celý obrázek, špatně adaptovatelná aplikace).

Většina SudokuNEKOM měla již předgenerovaná zadání a tento počet byl někdy dosti malý. V tomto případě mobilní aplikace ztrácí brzy na významu. Pokud SudokuNEKOM disponovalo generováním originálního zadání, trvalo generování většinou velmi dlouho. Například jedna SudokuNEKOM aplikace generovala novou Sudoku matici (bez rozlišení obtížnosti) na mobilním telefonu Sony Ericsson K750i (procesor 112MHz) cca 30 sekund. Generování bylo prováděno zřejmě pouhým backtrackingem, nebo s minimální aplikací heuristik.

SudokuKOM produkovala nová zadání relativně rychle. Čas tvorby nové Sudoku matice nebyl závislý na obtížnosti zadání, což je pro generátor nových originálních zadání atypické. Není zřejmé, zda obsahují předgenerovanou databázi Sudoku matic a nedochází k přímé generaci nové matice, jelikož se tyto Sudoku matice zdály být originální. Pokud je však počet předgenerovaných zadání značný, dochází k opakování zadání až po velmi dlouhé době a jeví se, jakoby generátor produkoval pořád nová jednoznačná zadání. Jedna z nejoblíbenějších a nejkomplexnějších SudokuKOM aplikací uvádí, že obsahuje 640 000 hlavolamů. Má tedy předgenerovaná zadání a aplikace si pamatuje (zřejmě ukládá do RMS), která byla uživateli nabídnuta, aby se zadání neopakovala v krátkém čase. Je možné, že i jiné SudokuKOM aplikace obsahují předgenerovaná zadání a další zadání si eventuálně generují a ukládají do paměti ve speciálním vlákne v době, kdy hráč řeší/hraje Sudoku matici (v této době nedochází k náročným výpočtům).

SudokuBP generuje vždy nové zadání, což je u požadavku na těžké zadání znát průměrně pětisekundovým zpožděním (viz *obrázek 3.3*). Generování je výpočetně relativně náročné a toto zpoždění by mohlo být u komerčních aplikací jistým nekomfortním omezením (obzvláště na mobilních telefonech s pomalejším procesorem).

Co se týče funkčnosti, aplikace SudokuBP je srovnatelná s komerčními aplikacemi. Všechny SudokuKOM obsahují generování nové hry rozlišené podle obtížnosti a některé automatický řešitel zadání (solver) – čímž disponuje i SudokuBP. Další shodné vlastnosti například byly (ne všechny SudokuKOM aplikace však tyto vlastnosti mají) nápověda čísel, ukládání rozehrané hry a statistika úspěšných řešitelů. Některé SudokuKOM byly doprovázeny zvukem, čímž SudokuBP nedisponuje. Podpora zvuku ve standartu MIDP 1.0 není, proto z důvodu přenositelnosti byla aplikace SudokuBP udělána bez zvuku.

SudokuNEKOM jsou funkčností daleko omezenější. Většinou obsahovaly pouhé generování nového zadání (někdy nerozlišovaly obtížnost), solver nebyl obsažen nikdy. Z běžných vlastností byla obsažena nápověda čísla, kontrola správnosti řešení, někdy statistika úspěšných řešitelů.

Významnější vlastnost, která nebyla nalezena u žádné cizí Sudoku aplikace, bylo importování Sudoku zadání z externího zdroje a kumulace výsledků na tomto externím zdroji. V SudokuBP se jedná o webovou aplikaci.

Souhrnné srovnávací informace se nachází na následujícím *obrázku 5.1 / tabulce*.

Aplikace	Hlavní menu	Herní plocha	Generování zadání	Solver	Zvuk	Statistika řešitelů
SudokuBP	seznam položek	obrázek, geometrická primitiva (přizpůsobení)	3 stupně obtížnosti, generátor popsáný v BP	ANO	NE	ANO
Nekomerční 1	seznam položek	geometrická primitiva (GP)	nerozlišuje obtížnost (náhodná zadání), někdy se generuje cca 30 sekund	NE	NE	ANO
Nekomerční 2	není, hned se zobrazí herní Sudoku matice	GP, na pozadí obrázek (nepřizpůsobuje rozlišení)	předgenerovaná zadání (pevný neznámý počet)	NE	NE	NE
Nekomerční 3	seznam položek	GP, na pozadí obrázek (nepřizpůsobuje rozlišení)	předgenerovaná zadání (pevný neznámý počet)	NE	NE	NE
Nekomerční 4	seznam položek, na pozadí obrázek	geometrická primitiva	předgenerovaná zadání (10 zadání)	NE	NE	NE
Nekomerční 5	seznam položek	obrázek-při nižším rozlišení nepřizpůsobuje	nerozlišuje obtížnost (náhodná zadání), generace cca 10 sekund	NE	NE	ANO
Komerční 1	obrázkové menu	obrázek pro konkrétní rozlišení (při stahování)	5 stupňů obtížnosti, 640 000 zadání	ANO	ANO	ANO
Komerční 2	obrázkové menu	obrázek pro konkrétní rozlišení (při stahování)	3 stupně obtížnosti; není známo, zda jsou zadání předgenerována	ANO	ANO	ANO
Komerční 3	obrázkové menu	obrázek pro konkrétní rozlišení (při stahování)	3 stupně obtížnosti; není známo, zda jsou zadání předgenerována	NE	NE	ANO
Komerční 4	geometrická primitiva, na pozadí obrázek	geometrická primitiva	předgenerovaná zadání, pro nová zadání nutné stáhnout aplikaci v jiné verzi	NE	NE	NE

Obrázek 5.1: Srovnání komerčních a nekomerčních Sudoku aplikací s aplikací vytvořenou v rámci bakalářské práce.

6 Závěr

Cílem bakalářské práce byla tvorba mobilní aplikace Sudoku se zaměřením na generické použití, tedy na nejširší okruh mobilních telefonů s podporou aplikační platformy J2ME. Jednotlivé aspekty tvorby aplikace byly odůvodněny s nástinem případné realizace jinými postupy.

Aplikace generuje zadání, která jsou řešitelná na základě metod, které jsou lidé běžně zvyklí používat. Nepochází tedy ke generaci náhodných zadání, která jsou řešitelná náhodnými, vůči obtížnosti nekompetentními, metodami. Generování Sudoku zadání je pro technologii mobilních telefonů výpočetně náročné a byly implementovány a otestovány dva přístupy generování, které se v praxi většinou používají. Výsledný algoritmus generování, jenž je součástí konečné aplikace Sudoku, dává relativně obstojné výsledky. Pro pohodlí uživatele je na místě algoritmus v rámci možností ještě zefektivnit, zvláště pak pro starší mobilní telefony s pomalejšími procesory.

Bakalářská práce si kladla za dílčí cíl disponovat vlastností, která je na trhu mobilních aplikací Sudoku jedinečná, a to možností přes síť Internet importovat zadání z webového serveru a zpětnou kumulaci výsledků úspěšných řešitelů. Je možné, že kdyby se trh obohatil o aplikaci disponující touto vlastností, přineslo by to zajímavou poptávku ze strany uživatelů.

Na doporučení kolegů se případné rozšíření aplikace do budoucna jeví implementovat možnost výměny Sudoku zadání pomocí technologie Bluetooth. Uživatel by si opsal a uložil jemu připadající zajímavé zadání z externího zdroje a po čase jinému vlastníkovu této aplikace Sudoku za pomoci Bluetooth nahrál do mobilního telefonu. Uživatel by potom nahrané zadání řešil.

Dalším možným vývojem projektu by mohlo být generování zadání pomocí genetických algoritmů. Jedná se spíše o experimentální záležitost, v praktickém nasazení zřejmě neobstojné.

Vytvořená aplikace je již rozšířena na nekomerční sféře mezi kolegy a s nemalým zájmem úspěšně v praxi využívána.

Literatura

- [1] Mahmoud, Q., H.: Naučte se Java 2 Micro Edition. Nakladatelství Grada Publishing, a.s., Praha, 2002. Přel. z Learning Wireless Java. ISBN: 80-247-0444-7.
- [2] Piroumian, V.: Wireless J2ME™ Platform Programming. Prentice Hall PTR, 2002. ISBN: 0-13-044914-8.
- [3] Spell, B.: Java programujeme profesionálně. Nakladatelství Computer Press, a.s., Brno, 2002. Přel. z Professional Java Programming. ISBN: 80-7226-667-5.
- [4] Herout, P.: Učebnice jazyka Java. Nakladatelství Kopp, České Budějovice, 2000. ISBN: 80-7232-115-3.
- [5] J2ME™ MIDP Specification 1.0a Final Release. Java Sun Technology [online]. 2000-12-15 [cit. 2007-04-14]. Dostupné z WWW: <http://192.18.108.238/ECom/EComTicketServlet/BEGINF7CD90E33156CC1F8C86529C3B9964B6/-2147483648/2110870443/1/379754/379742/2110870443/2ts+/westCoastFSEND/7580-j2me_midp-1.0a-fr-spec-oth-JSpec/7580-j2me_midp-1.0a-fr-spec-oth-JSpec:1/midp-1_0a-spec.zip>
- [6] Sun Java Wireless Toolkit 2.5. Java Sun Technology [online]. 2007-01-29 [cit. 2007-03-06]. Dostupné z WWW: <http://java.sun.com/products/sjwtoolkit/download-2_5.html>
- [7] Hodek, V. Programujeme v J2ME. PCSvět [online]. 2004-03-22 [cit. 2007-04-13]. Dostupné z WWW: <<http://www.pcsvet.cz/java/>>
- [8] Hodek, V. Fórum pro vývojáře mobilních aplikací. vhodek [online]. [cit. 2007-04-15]. Dostupné z WWW: <<http://www.j2me.vhodek.net/index.php>>
- [9] Bittnerová, L., R., Procházka, J., Kovařík, P. Vývoj aplikací v J2ME. Interval [online]. 2005-06-13 [cit. 2007-04-13]. Dostupné z WWW: <<http://interval.cz/vyvoj-aplikaci/j2me/>>
- [10] The Java ME Device Table. Java Sun Technology [online]. [cit. 2007-04-17]. Dostupné z WWW: <<http://developers.sun.com/techtopics/mobility/device/device>>
- [11] Sudoku, *Wikipedia* [online]. 2007-05-06 [cit. 2007-05-06]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Sudoku>>
- [12] Mathematics of Sudoku, *Wikipedia* [online]. 2007-04-24 [cit. 2007-04-27]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Mathematics_of_Sudoku>
- [13] Tomlinson, H. Techniques For Solving Sudoku. PalmSudoku [online]. 2005 [cit. 2007-04-23]. Dostupné z WWW: <<http://www.palmsudoku.com/pages/techniques-overview.php>>
- [14] How to generate random Sudoku array. Sudoku Programmers Forum [online]. 2006-10-03 [cit. 2007-04-24]. Dostupné z WWW: <<http://www.setbb.com/phpbb/viewtopic.php?t=314&mforum=sudoku>>

- [15] NP-úplnosť, *Wikipedia* [online]. 2007-04-28 [cit. 2007-04-28].
Dostupné z WWW: <<http://cs.wikipedia.org/wiki/NP-%C3%BAplnost>>
- [16] Exact cover, *Wikipedia* [online]. 2007-03-14 [cit. 2007-04-27].
Dostupné z WWW: <http://en.wikipedia.org/wiki/Exact_cover>
- [17] Knuth, D., E. Dancing Links, Stanford University. arXiv [online].
2000-11-15 [cit. 2007-04-27]. Dostupné z WWW:
<http://xxx.lanl.gov/PS_cache/cs/pdf/0011/0011047.pdf>
- [18] Sudoku solver using Knuth's "Dancing Links" method. Sudoku Programmers Forum [online].
2007-01-16 [cit. 2007-04-26]. Dostupné z WWW:
<<http://www.setbb.com/phpbb/viewtopic.php?t=605&mforum=sudoku>>

Seznam příloh

Příloha A - Manuál použití aplikace

Příloha B - CD (zdrojové kódy, programová dokumentace, technická zpráva, vývojové prostředí)

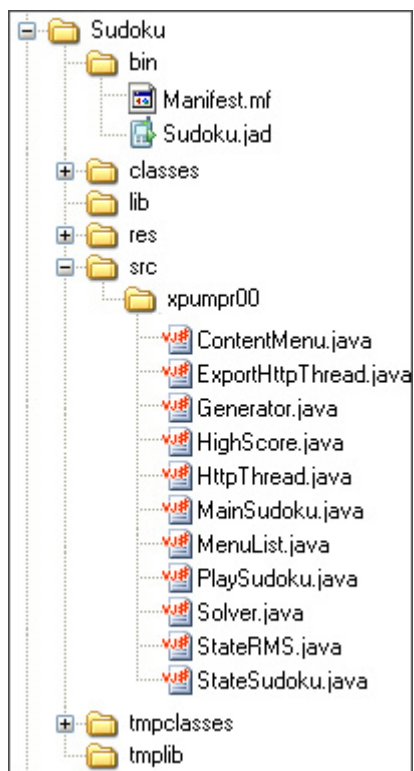
Příloha A - Manuál použití aplikace

Kompilace

Pro kompilaci aplikace je silně doporučeno používat integrované vývojové prostředí *J2ME Wireless Toolkit*, ve kterém se dá aplikace taktéž emulovat (emulace mobilního telefonu). Nejnovější instalační balík je zdarma k dostání na oficiálních stránkách Javy [6]. Toto programové vybavení ve verzi 2.5 (pro operační systém Windows, datum zveřejnění této verze - 29.1.2007) je též k nalezení na přiloženém CD nosiči (soubor `sun_java_wireless_toolkit-2_5-windows.exe`), který je součástí bakalářské práce.

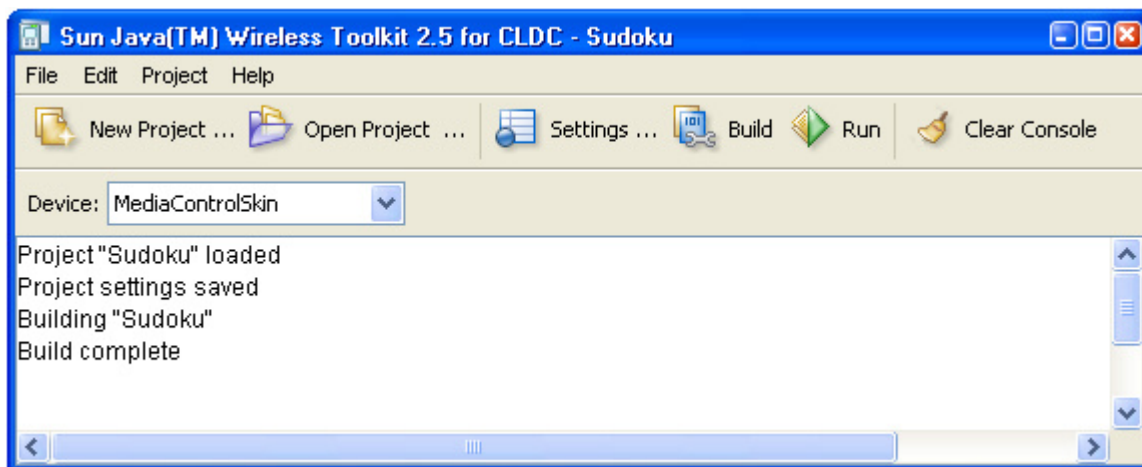
Po instalaci vývojového prostředí se překopíruje adresář *Sudoku* (je na přiloženém CD) do adresáře *apps*, který je součástí adresáře *WTK25* (Adresář *WTK25* byl vytvořen po instalaci vývojového prostředí.). V adresáři *apps* se nachází veškeré naprogramované aplikace / projekty určené pro kompilaci / spuštění.

Adresář *Sudoku* obsahuje vše potřebné pro úspěšnou kompilaci. Jedná se o zdrojové soubory (`./Sudoku/src/xpump00/*.java`); obrázky použité v aplikaci (`./res/.`); JAD soubor – Java Application Descriptor (`./Sudoku/bin/Sudoku.jad`) a soubor Manifestu (`./Sudoku/bin/Manifest.mf`). Adresářová struktura je zobrazena na obrázku A.1.



Obrázek A.1: Adresářová struktura projektu Sudoku.

Ve vývojovém prostředí *J2ME Wireless Toolkit* (obrázek A.2) se otevře/vybere projekt poklepáním na tlačítko *Open Project*. Zde se ze seznamu různých projektů vybere projekt s názvem *Sudoku* (je-li již ve zmiňovaném adresáři *apps*, jinak není v seznamu projektů zobrazen).



Obrázek A.2: Vývojové prostředí *J2ME Wireless Toolkit 2.5*.

Nyní se zmáčkne tlačítko *Build* (obrázek A.2), čímž dojde ke kompilaci projektu *Sudoku* (vytvoří se příslušné *.class soubory). V tomto stavu je aplikace již spustitelná/emulovatelná ve vývojovém prostředí *J2ME Wireless Toolkit*.

Pokud je cílovým zařízením mobilní telefon, je ještě nutné vytvořit *jar* soubor (*Sudoku.jar*). Jedná se o zkomprimovaný soubor, který obsahuje přeložené *.class soubory, obrázky, manifestový soubor a jiné podpůrné soubory, které jsou nutné k chodu aplikace. Výsledný *jar* soubor se vytvoří v *J2ME Wireless Toolkit* následovně: *Project* → *Package* → *Create Package*. Nyní je *Sudoku.jar* vytvořen (nachází se v *./Sudoku/bin/Sudoku.jar*) a může se nahrát do mobilního telefonu.

Další podkapitoly budou popisovat aplikaci z pohledu uživatele, tedy funkčnost a použití aplikace. V *J2ME Wireless Toolkit* se aplikace spouští (pokud je vše úspěšně zkompilováno) poklepáním na tlačítko *Run* (obrázek A.2).

Menu

Jakmile se aplikace spustí, je zobrazeno hlavní menu (obrázek 3-1).

V menu lze vybrat následující položky:

- **Play** (generace nové hry *Sudoku*)
- **Solver** (automatické dořešení *Sudoku* zadání)
- **Load Game** (dohrání rozehrané uložené hry)
- **Import** (Import zadání z webového serveru)
- **Settings** (nastavení vlastností hry)
- **HighScore** (časová statistika úspěšnosti hráčů)

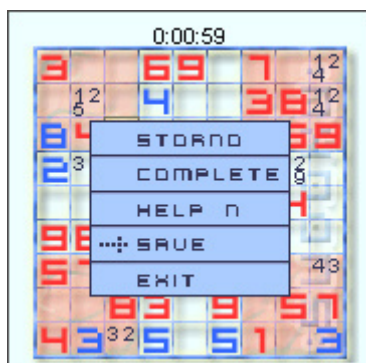
- **Help** (pravidla řešení hry Sudoku a vysvětlení ovládání hry)
- **Author** (informace o autorovi aplikace)
- **Exit** (ukončení aplikace)

Play

Dochází ke generování Sudoku zadání dle zvolené obtížnosti. Obtížnost se nastavuje v **Settings** (easy - lehká, intermediate - střední, hard - těžká).

Jakmile je zadání vygenerováno, může uživatel řešit (obrázek 3.4). Předgenerovaná čísla jsou barvou červenou a přirozeně není možné je modifikovat. Uživatелеm zadaná čísla mají barvu modrou. Není možné zadat číslo, které je kolizní (ve smyslu Sudoku podmínek) s předgenerovanými čísly. Je možné si však dělat do políček poznámky – do políčka až čtyři čísla (kandidáty, tzv. pencil-marks). Pokud je však aplikace spuštěna na mobilním telefonu s rozlišením menším než 170 x 170px (obrázek 3.5), je možné do políčka zadat maximálně jedno číslo (nelze si tedy poznamenat více kandidátů). Vepsání více čísel do políčka vede ke špatnému rozeznání (prakticky nerozeznatelné). Čísla se zadávají numerickými tlačítky, mazání se provádí nulou.

Během hry lze vybrat kontextové herní menu (obrázek A.3). Herní menu se vyvolává tlačítkem s mřížkou (vězením).



Obrázek A.3: Herní menu.

V herním menu lze vybrat položky: **Storno**, **Complete**, **Help N**, **Save**, **Exit**.

Storno zapříčiní zrušení herního menu.

Complete ověřuje, zda je matice úspěšně vyřešena. Pokud matice není úspěšně vyřešena, pokračuje se ve hře (herní menu se odkrývá). Pokud je matice správně vyplněna, uživatel zadává svoje jméno a dochází k uložení tohoto jména a času řešení do seznamu úspěšných řešitelů (k příslušné obtížnosti).

Help N napovídá / odkrývá uživateli na zvolených místech správná čísla. Počet napovídajících čísel lze měnit v nastavení (1 až 6 čísel nápovědy). Pokud uživatel použije nápovědu, nedochází po dořešení Sudoku matice k ukládání výsledku do seznamu úspěšných řešitelů.

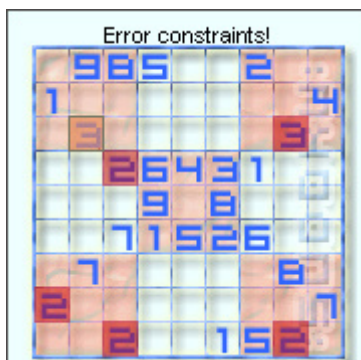
Save slouží k ukládání rozehraného stavu hry. Rozehraná matice se ukládá pod uživatelsky zvoleným názvem. Tento název musí být jedinečný (nemůže existovat uložená matice pod stejným názvem).

Exit ukončuje rozehranou hru a návrat do hlavního menu.

Solver

Solver slouží k automatickému dořešení Sudoku matice.

Uživatel vepíše do prázdných polí čísla (opět numerická čísla, nula maže), zmáčkne tlačítko mřížka (vězení) a v kontextovém menu vybere položku **Solve**. Jestliže uživatelem zadaná čísla jsou v rozporu se Sudoku podmínkami, dojde k označení chyb (obrázek A.4). Pokud matice řešení nemá, vypíše se, že matice nelze dořešit. V případě, že má matice více řešení (což ale není správně zadaná Sudoku matice), je zobrazeno první nalezené řešení.



Obrázek A.4: Solver, zadaná čísla jsou v rozporu se Sudoku podmínkami.

Load game

Uživatel má možnost dohrát již v minulosti uloženou hru. Vybírá se ze seznamu názvů uložených stavů hry. Pokud se některý uložený stav vybere, dochází ke smazání tohoto stavu ze seznamu a uživatel pokračuje v znovuustavené rozehrané hře.

Import

Další vlastností aplikace je možnost si importovat zadání z webového serveru. Dochází k navázání komunikace s webovým serverem a následné importování zadání. Pro tento účel je nutné, aby měl uživatel na svém mobilním telefonu nastavený přístupový bod (Access Point) na Internet, nikoliv na WAP (komunikace totiž probíhá pomocí HTTP protokolu).

Pro daný týden/rok existuje pouze jedno zadání. Tedy všichni uživatelé, kteří v daném období požádají o import, dostanou stejnou Sudoku matici k řešení. Pokud pro určitý týden/rok neexistoval

někdo, kdo by podal požadavek na import, nedochází v tomto období ke generaci zadání (bylo by to bezúčelné).

Ve fázi, kdy uživatel řeší importovanou Sudoku matici, je obvyklá funkčnost omezena. Lze buď zrušit rozehranou hru, nebo potvrdit správnost řešení (nelze tedy využít nápovědu, nebo si rozehranou hru uložit). Pokud je řešení správné, je uživatel vyzván k zadání svého jména. Jméno řešitele a čas řešení se odesílá na webový server, kde se nachází statistika úspěšných řešitelů.

Stránky s výsledky jsou na adrese <http://www.tsholesov.cz/sudoku/>. Vlevo se nachází menu, kde lze zvolit příslušný týden/rok. Pod každým týdnem/roce se zobrazuje statistika úspěšných řešitelů (obrázek A.5).

rok 2007, 18. týden			
Časy řešitelů:			
pořadí	jméno řešitele	čas řešení	datum vyřešení
1.	Pumprla	1:17	2007-04-30 10:29:36
2.	Ondra	2:24	2007-05-06 14:26:10
3.	Mr. Kennyhack	2:27	2007-04-30 10:26:50
4.	Petr	2:43	2007-05-01 23:55:23
5.	mr. Zkouska	2:45	2007-05-03 09:32:37

Obrázek A.5: Statistika úspěšných řešitelů pro daný týden / rok na webových stránkách.

Pokud je daný týden uplynulým, zobrazí se taktéž vyřešená Sudoku matice, která byla určena pro toto období.

Na stránkách lze u každého týdnu/roku přidávat komentáře (např. reakce na obtížnost zadání, apod.). Vkládání komentářů je ošetřeno proti automatickým softwarovým robotům.

Settings

Jedná se o nastavení některých herních vlastností.

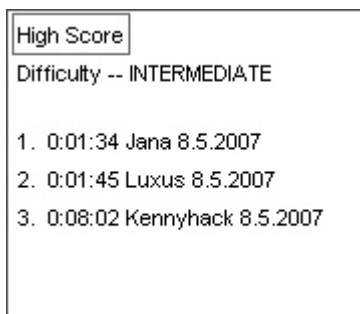
Lze nastavit obtížnost generování zadání – obtížnosti jsou charakteristické metodami, které musí být pro vyřešení Sudoku matice aplikovány, viz kapitola 3.2.1.1 *Generování zadání*.

Dále může uživatel nastavit počet políček, které mohou být odkryty - nápověda.

Poslední možností v nastavení je časové omezení hry. Uživatel může zvolit maximální čas, který bude mít na vyřešení Sudoku matice. Po tomto čase hra automaticky končí.

Highscore

Po zvolení tohoto políčka v hlavním menu se zobrazí seznam úspěšných řešitelů. Zobrazí se seznam příslušející k takové obtížnosti, jaká je zvolena v nastavení (settings). V tomto seznamu je zobrazeno pořadí, jméno řešitele, čas řešení a datum, kdy bylo daného skóre dosaženo (obrázek A.6).



Obrázek A.6: Seznam úspěšných řešitelů (Highscore).

Help

Popis Sudoku pravidel a ovládání aplikace.

Author

Kontakt na autora aplikace.

Exit

Dochází k ukončení aplikace.