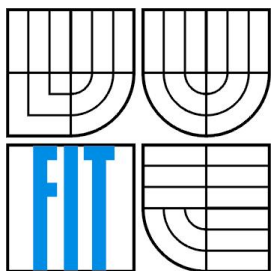




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# KARETNÍ HRA MARIÁŠ PRO MOBILNÍ ZAŘÍZENÍ

CARD GAME MARIÁŠ FOR MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

Tomáš Pumprla

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Zbyněk Křivka, Ph.D.

BRNO 2007

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Pumpřila Tomáš**

Obor: Informační technologie

Téma: **Karetní hra Maryáš pro mobilní zařízení**

Kategorie: Uživatelská rozhraní

#### Pokyny:

1. Nastudujte pravidla karetní hry Maryáš a vyberte vhodné implementační prostředí pro grafickou aplikaci zajišťující hru jak proti lidem, tak proti počítačovým protivníkům.
2. Nastudujte některou z vhodných metod umělé inteligence pro hru počítače (např. expertní systém) a výběr metody i prostředí v práci i s vedoucím podrobně diskutujte.
3. Hru implementujte s možností dále vylepšovat inteligenci počítačových protihráčů.
4. Implementaci otestujte, porovnejte alespoň s jednou existující implementací a navrhnete budoucí vylepšení.

#### Literatura:

- Moravcová, J.: Deskové a karetní hry, Grada, Praha, 2005, 122 s., ISBN: 80-247-1339-X
- Russell, Stuart J.: Artificial intelligence: A Modern Approach, Prentice-Hall, New Jersey, 2003, 1081 s.
- dokumentace s použitím vývojového prostředí a mobilnímu zařízení

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

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 nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude uloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Křivka Zbyněk, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2

---

doc. Ing. Jaroslav Zendulka, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA  
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Tomáš Pumprla**  
Id studenta: 79297  
Bytem: Sadová 1530, 769 01 Holešov  
Narozen: 03. 02. 1986, Kroměříž  
(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: Karetní hra Maryáš pro mobilní zařízení  
Vedoucí/školitel VŠKP: Křivka Zbyněk, Ing., Ph.D.  
Ústav: Ústav informačních systémů  
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 zpracování 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

.....

*Ogler 2015*  
Autor

## **Abstrakt**

Tato práce se zabývá vývojem mobilní aplikace založené na programovacím jazyce Java - J2ME. Aplikací je karetní hra Mariáš, přičemž uživatel může zvolit hru proti lidem nebo proti počítači, který se rozhoduje na základě expertního systému. Expertní systém je možné aktualizovat prostřednictvím HTTP komunikace.

## **Klíčová slova**

J2ME, MIDP, CLDC, RMS, MIDlet, HTTP, generická mobilní aplikace, expertní systém, J2ME Wireless Toolkit

## **Abstract**

This thesis deals with development of application for mobile phones programmed in Java - J2ME. The application is a card game Marias. User can choose play with two other people or with computer which is managed by expert system. This expert system is able to update its "knowledgebase" via HTTP communication.

## **Keywords**

J2ME, MIDP, CLDC, RMS, MIDlet, HTTP, generic mobile application, exper system, J2ME Wireless Toolkit

## **Citace**

Pumpla Tomáš: Karetní hra Mariáš pro mobilní zařízení. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Karetní hra Mariáš pro mobilní zařízení

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zbyňka Křivky, Ph.D.

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

.....  
Tomáš Pumprla  
14. května 2008

## Poděkování

Děkuji svému vedoucímu Ing. Zbyňkovi Křivkovi, Ph.D. za jeho rady, připomínky a čas, který mi věnoval. Děkuji svým rodičům, kteří mě podporovali a umožnili mi studovat a dostudovat.

Děkuji Mirečece za vše, čím pro mě je.

© Tomáš Pumprla, 2008.

*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 .....	5
1.1 Úvod do J2ME .....	5
1.2 Konfigurace .....	6
1.2.1 CLDC konfigurace .....	7
1.2.2 CDC Konfigurace .....	8
1.3 Profily .....	8
1.3.1 MIDP 1.0 .....	9
1.3.2 MIDP 2.0 .....	9
1.4 MIDlet .....	10
2 Karetní hra Mariáš .....	11
2.1 Obecná pravidla karetní hry Mariáš .....	11
2.2 Typy her .....	12
3 Expertní systémy .....	13
3.1 Obecné vlastnosti Expertních systémů .....	13
3.1.1 Znalostní báze .....	13
3.1.2 Řídící mechanismus .....	13
3.2 Klasifikace expertních systémů na základě strategie procesu usuzování .....	14
3.2.1 Diagnostický expertní systém .....	14
3.2.2 Plánovací expertní systém .....	15
3.3 Expertní systém v aplikaci Mariáš .....	16
4 Midlet Mariáš .....	17
4.1 Předběžné aspekty před tvorbou aplikace .....	17
4.2 Uživatelské rozhraní .....	18
4.3 Ukládání rozehrané hry .....	20
4.4 Aktualizace UI z webového serveru .....	21
4.5 Settings, Help, Credits, Edit names .....	21
5 Webové systém pro bezplatnou distribuci mobilních aplikací .....	22
5.1 Účel webových stránek .....	22
5.2 Funkčnost webových stránek .....	22
6 Porovnání s cizími aplikacemi .....	23
7 Závěr .....	24
Literatura .....	25

Seznam příloh .....	26
Příloha 1 – Manuál použití aplikace .....	27
Kompilace .....	27
Menu.....	28
Play.....	29
Settings .....	30
Load saved game .....	31
Help .....	31
Credits .....	31
Exit .....	31



# Úvod

Dnes, v době, kdy každá vteřina hraje svou roli, v době, kdy se svět neobejde bez technologických vymožeností dneška a zítra bez technologie zítřka, je takřka každý člověk závislý na mobilní komunikaci a s ní spojeným mobilním telefonem. V současnosti kvůli víceúčelovosti podporuje většina těchto telefonů aplikační technologii J2ME (Java 2, Micro Edition), v níž je vytvořeno mnoho užitečných či zábavných programů, které mají za úkol uživateli zpříjemnit práci či danou chvíli.

Tato bakalářská práce je zaměřena právě na druhou skupinu programů, určených k pobavení, konkrétněji – vytvořením karetní hry **Mariáš**, která doposud na mobilních telefonech nebyla realizována. Z důvodu snahy vytvořit nejatraktivnější aplikaci pro uživatele, bylo snahou využít co nejvíce možností platformy J2ME se zachováním, pokud možno, největší přenositelnosti mezi přístroji umožňující podporu Javy, což může být občas problém vzhledem k různým verzím konfigurací a profilů J2ME (viz Kapitola 1.2 *Konfigurace*, 1.3 *Profily*), velikostem displeje nebo velikosti paměti pro běhové prostředí Javy.

Při hraní proti počítači ovládanému umělou inteligencí je občas vhodné tuto inteligenci změnit, zpravidla k lepšímu. Aby uživatelé kvůli vylepšení nemuseli nahrávat celou aplikaci do mobilního telefonu znovu, mají možnost stáhnout si aktualizaci pravidel expertního systému z webového serveru prostřednictvím protokolu HTTP přímo z aplikace.

Aplikace je určená pro mobilní telefony s podporou J2ME (Java 2, Micro Edition) s barevným displejem, CLDC 1.0 konfigurací a MIDP 2.0 profilem, čímž je zaručena relativně velká přenositelnost. Pro převedení aplikace na neomezeně přenositelnou, neboli na tzv. generickou aplikaci, by bylo nutné upustit od profilu MIDP 2.0, který nezvládají některé starší mobilní telefony s platformou J2ME a přejít na profil MIDP 1.0, od něhož se aplikace liší jen celoobrazovým módem full screen. Důvodem rozhodnutí použít celoobrazového módu bylo zvážení možných velikostí displejů a velikosti (čitelnosti) zobrazovaného obsahu.

Pro dostupnost této aplikace byly vytvořeny speciální webové stránky, napsané v jazycích HTML, CSS, PHP a MySQL, jejichž účelem je distribuce volně šířitelných autorských mobilních aplikací, na nichž může každý vývojář nabídnout své programy širší veřejnosti, případně jen dalším vývojářům na ukázkou či otestování.

Aplikace **Mariáš** byla v průběhu testována (emulována) a odladována v *Java™ Wireless Toolkit 2.5 for CLDC*, jenž je zdarma k dostání na oficiálních stránkách Java Sun [3]. V praktickém nasazení pak byla testována převážně na mobilním telefonu Sony Ericsson K750i.

Jedná se o samostatný projekt, který nenavazuje na žádnou předešlou cizí práci.

V následujícím textu se bakalářská práce zabývá obecnými pravidly karetní hry Mariáš, následně pak technologií J2ME, a to hlavně obecným popisem a souvislostmi s vývojem konkrétních mobilních aplikací. Další kapitola popisuje expertní systémy, za jejichž využití byla implementována

umělá inteligence počítače. Následující kapitola podrobněji popisuje implementaci některých vlastností mobilní aplikace **Mariáš** a jakými technologiemi bylo toho dosaženo. Pátá kapitola pojednává o webových stránkách, které slouží k distribuci volně šiřitelných autorských mobilních aplikací. Poslední kapitolou je srovnání aplikace Mariáš v rámci bakalářské práce s jinými aplikacemi.

# 1 Technologie J2ME

Aplikace karetní hra **Mariáš** pro mobilní zařízení byla implementována v jazyce J2ME a právě tato kapitola se zabývá technologií J2ME. Jsou zde popisovány jednotlivé důležité aspekty této programovací platformy s důrazem na popis různých používaných standardů.

## 1.1 Úvod do J2ME

Trend expanzivně rostoucí dostupnosti malých přenosných zařízení si vyžádal tvorbu relativně robustního programového prostředí pro tato malá zařízení. Reakcí na výzvu tvorby účelného a přitom nenáročného programového prostředí se úspěšně chopila firma Sun. Zpočátku vytvořila několik platforem založených na JDK 1.1 (což je předchůdce platformy Java 2). V roce 2001 společnost Sun vystoupila na konferenci JavaOne a představila J2ME, čímž měla nahradit produkty na bázi JDK 1.1 unifikovanějším řešením na bázi Java 2.

Společnost Sun doposud vyvinula tři platformy jazyka Java, které se různí (ve smyslu funkčnosti a hardware výkonností) svými požadavky na cílový systém, pro které jsou tyto programové platformy vyvíjeny. Mezi tyto platformy patří:

- **J2EE** – Java 2 Enterprise Edition (serverové orientované aplikace)
- **J2SE** – Java 2 Standart Edition (desktopově orientované aplikace)
- **J2ME** – Java 2 Micro Edition (aplikace pro přenosná zařízení)

Právě J2ME (někdy též JME) platforma klade na cílový systém nejmenší nároky (hardwarové nároky budou popsány v kapitole *1.2 Konfigurace*).

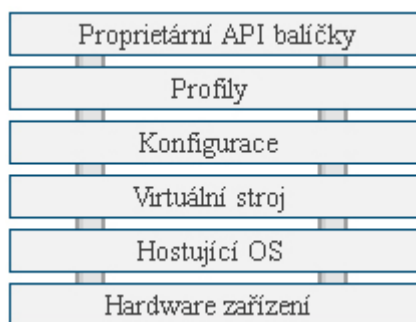
J2ME poskytuje kompletní řešení pro trh nejmodernějších síťových aplikací pro malá zařízení, jako jsou například mobilní telefony, osobní digitální asistenti (PDA), organizéři, pagery, navigační přístroje pro mobilní automobily a další malá přídavná zařízení. Výrobci, poskytovatelům a vývojářům těchto zařízení také slibuje možnost vyvíjet nové aplikace pro své zákazníky. Tato možnost však neznamena, že by byly obětovány základní vlastnosti Javy, jejichž význam v současnosti stále roste, jmenovitě kompatibilita různých prostředí a zabezpečení. Tento vývoj J2ME je zajištěn projektem zvaným Java Community Process (JCP) a od doby vzniku se k vývoji J2ME připojilo více než 600 společností, například Palm, Sony Ericsson, Nokia, Motorola a RIM. [1]

Platforma J2ME však není jediným standardem. Rozdílnost (použitelnost, výkonnost) přístrojů, pro které je určena, si vynutila další členění programového balíku J2ME. Například obecně mobilní telefony jsou hardwarově výkonnostně na nižší úrovni než PDA zařízení, a proto by nebylo vhodné použít stejný virtuální stroj J2ME pro obě množiny přístrojů. Stejná paralela lze nalézt u starších a novějších mobilních telefonů. J2ME je tedy kolekce technologií a specifikací, které ve vzájemné kombinaci tvoří *Java runtime* prostředí pro různé oblasti mobilních zařízení.

Platforma J2ME definuje následující komponenty [2]:

- *Virtuální stroj* specifický pro dané zařízení. Je primárně ovlivněno pamětí cílového zařízení.
- *Konfigurace*, které jsou určeny hardwarovým vybavením (procesor, paměť) zařízení. Zajištění základních knihoven. Podrobněji v *1.2 Konfigurace*.
- *Profily*, které vychází z užšího členění pro danou konfiguraci (konkrétní *konfigurace* je dále členěna na tyto *profily*). Dále rozšiřuje programové vybavení. Podrobněji v *1.3 Profily*.
- *Proprietární API* balíčky, které nemusí být přímo standardizovány (většinou specifické pro mobilní telefony konkrétní značky).
- Nástroje pro vývoj a nastavení zařízení (různé emulátory, například standardizovaný emulátor *Java Wireless Toolkit 2.5* [3]).

*Virtuální stroj*, *konfigurace* a *profily* pak tvoří tzv. *pracovní prostředí J2ME*. Jedná se o nezákladnější vrstvy, které je důležité při tvorbě aplikace uvažovat (zvláště pak *konfigurace* a *profily*). Následuje *obrázek 1.1* s hierarchií právě popsanych vrstev [1].



Obrázek 1.1: Hierarchie prostředí J2ME

## 1.2 Konfigurace

Tvůrci specifikace J2ME tvořili a tvoří jednotlivé standardy na základě společných rysů zařízení, pro které jsou určeny. Nejnižším společným jmenovatelem je hardware zařízení. Jsou to například procesory (instrukční sada, frekvence), velikost paměti (ROM a RAM), výdrž napájení a grafický výstup. Tyto aspekty jsou pak specifikovány konkrétní *konfigurací*. Zařízení, jenž splňují konkrétní *konfiguraci*, musí splňovat určité normy a toho programátor taktéž při tvorbě využívá (například některé *konfigurace* nespécifikují práci s plovoucí řádovou čárkou a není možné při programování tyto datové typy používat).

Konfigurace je určena následujícími aspekty [1]:

- podporované rysy virtuálního stroje
- podporované rysy programovacího jazyka Java (podmnožina J2SE)
- podporované základní javovské knihovny a API

V současné době jsou standardizovány dva typy konfigurací. Prvním typem je *CLDC konfigurace*, která je například určena pro všechny mobilní telefony a některá low-level PDA zařízení. Druhým a robustnějším typem je *CDC konfigurace*, která je určena pro většinu PDA zařízení, různé vestavěné systémy (*embedded systems*), apod.

## 1.2.1 CLDC konfigurace

*CLDC konfigurace* (*Connected Limited Device Configuration*) je určena pro výkonnostně relativně slabší zařízení, jako jsou například mobilní telefony, low-level PDA a pagery. Tato konfigurace používá tzv. *Kilo Virtual Machine*, což je skutečný virtuální stroj podle specifikace pro virtuální stroje Javy primárně určený pro zařízení s pamětí několika stovek kilobajt. V poslední době je tento virtuální stroj nahrazován *CLDC HotSpot Virtual Machine*, který při stejných paměťových nárocích dává až osmi násobné zrychlení [4]. Mobilní aplikace **Mariáš** používá právě tento typ konfigurace - *CLDC*.

Požadavky pro *J2ME CLDC* jsou následující [1,4]:

- Zařízení je řízeno 16-bitovým nebo 32-bitovým procesorem s taktovací frekvencí minimálně 25MHz (velmi často je tato frekvence 50 MHz až 200Mhz)
- Minimální velikost paměti RAM je 300KB (pro běh aplikací), minimální velikost ROM a flash je 1MB (statická paměť). Tyto hodnoty nejsou však vždy striktně dodrženy (různé zdroje taktéž uvádí různé hodnoty; uvedené hodnoty jsou ze [4]).
- Zařízení může mít omezený zdroj energie (často baterie)
- Zařízení je propojitelné s některým typem sítě často bezdrátovým s přerušovaným připojením a s omezenou šířkou pásma (je například nutná implementace protokolu HTTP).

Zařízení s *CLDC* konfigurací definují následující programové rozhraní (následující balíčky odpovídají balíčkům edice J2SE, jsou však ochuzeny o některé třídy) [5]:

- `java.lang`
- `java.io`
- `java.util`
- `javax.microedition.io` (oproti J2SE nová knihovna pro I/O operace)

V rámci *CLDC konfigurace* se dále ještě rozlišuje verze *CLDC 1.0* a *CLDC 1.1*. Zařízení splňující *CLDC 1.0* nepoužívají plovoucí řádovou čárku. Nelze tedy používat datové typy jako `float` nebo `double`. Zařízení splňující normu *CLDC 1.1* jsou již založeny na plovoucí řádové čárce a lze tedy využívat příslušných datových typů (`float` a `double`). Mobilní aplikace **Mariáš** používá konfiguraci *CLDC 1.0*, jelikož není třeba využívat desetinných čísel.

## 1.2.2 CDC Konfigurace

*Konfigurace CDC (Connected Device Configuration)* je používána oproti konfigurace CLDC inteligentnějšími zařízeními. Mezi tato zařízení patří většina PDA, navigační přístroje pro automobily, chytré komunikátory (MDA), televize po Internetu, apod. CDC obsahuje plnou verzi virtuálního stroje Javy, který se dnes používá ve J2SE [1]. Tato konfigurace se v mobilních telefonech dnes běžně ještě nepoužívá, ačkoliv některé mobilní telefony svými vlastnostmi splňují většinu bodů ze specifikace CDC.

Požadavky pro *J2ME CDC* jsou následující [6]:

- Zařízení je řízeno 32-bitovým procesorem (frekvence nespécifikována).
- Pro běhové prostředí Javy zařízení disponuje 2 MB (RAM + ROM).
- Zařízení vyžaduje plně funkční virtuální stroj Java 2 „*Blue Book*“.
- Zařízení je propojitelné s některým typem sítě často bezdrátovým s přerušovaným připojením a s omezenou šířkou pásma.

Zařízení s CDC konfigurací definují následující programové rozhraní, které je nadmnožinou programového rozhraní CLDC [5]:

- `java.net`
- `java.text`
- `java.security`

## 1.3 Profily

Jak již bylo zmíněno v článku 1.1 *Úvod do J2ME*, v rámci jedné konfigurace se dále rozlišuje mezi více *profily*. Pro programátora mobilních aplikací (speciálně mobilních telefonů) je tato vrstva nejzásadnější. Zatímco v rámci konfigurace se při tvorbě aplikací pro mobilní telefony používá zásadně CLDC konfigurace (a rozhoduje se pouze na základě použití desetinných čísel, CLDC 1.0 nebo CLDC 1.1), volba profilu tak jednoznačná není.

*Profil* určuje sadu programových rozhraní (API – třídy pro uživatelská rozhraní, trvalé ukládání dat, práce v síti), která je nadstavbou konfigurace. Členění různých typů profilů je daleko markantnější než členění konfigurací. Dochází již k užší specifikaci pro danou množinu mobilních zařízení. Pro mobilní telefony se nejčastěji používá profily MIDP 1.0 a MIDP 2.0, který dále rozšiřuje profil MIDP 1.0. Někdy se komerčně u aplikací uvádí označení *Java 1* pro MIDP 1.0 a *Java 2* pro MIDP 2.0 (tyto komerční názvy jsou však dosti zavádějící). Výběr profilu MIDP 1.0 či MIDP 2.0 je dosti zásadní, jelikož implikuje robustnost přenositelnosti. Zatímco všechny mobilní telefony splňují MIDP 1.0, u MIDP 2.0 tomu tak není. Je však pravdou, že drtivá většina mobilních telefonů posledních let již MIDP 2.0 splňuje [7]. Mobilní aplikace **Mariáš** je programována v rámci profilu

*MIDP 2.0*, a to z důvodu běhu aplikace v celoobrazovém módu (*fullscreen mode* není v *MIDP 1.0* obecně podporován a bylo by třeba sáhnout po proprietárním řešení, čímž by se přenositelnost značně snížila).

### 1.3.1 MIDP 1.0

*MIDP 1.0* je základním profilem pro mobilní telefony a musí být splněna všemi zařízeními, které disponují edicí J2ME. (Praxe je však z osobní zkušenosti jiná a některé standardy dodrženy nejsou!). *MIDP 1.0* je spravován komunitou *Java Community Process* (JSR 37 [8]).

Aspekty *MIDP 1.0* jsou následující [1,8]:

- Velikost obrazovky musí mít minimálně rozlišení 96 x 54 pixelů. Minimální hloubka barev 1 bit.
- Zařízení musí mít „klávesnici pro jednu ruku“ (běžné mobilní telefony), „dvojrúční klávesnici“ (tzv. QWERTY), nebo dotykovou obrazovku.
- 32 KB nestálé paměti pro práci Javy (JRE, *Java heap*)
- 128 KB stálé paměti pro komponenty *MIDP*
- 8 KB stálé paměti pro dlouhodobé ukládání dat z aplikací (např. pomocí *RMS*)
- Přerušované obousměrné spojení, obvykle bezdrátové s omezenou šířkou pásma

*MIDP 1.0* rozšiřuje programové rozhraní *CLDC* o následující balíčky [1]:

- `javax.microedition.midlet` – plánovač běhu aplikace (viz Kapitola 1.4 *MIDlet*).
- `javax.microedition.lcdui` – tvorba grafického uživatelského rozhraní
- `javax.microedition.rms` – ukládání dat v databázovém záznamovém systému
- Rozšíření sady rozhraní `ContentConnection` rozhraním `HttpConnection`, čímž je umožněna HTTP komunikace.

### 1.3.2 MIDP 2.0

Abstrahování některých vlastností v *MIDP 1.0* (například pokud měla být aplikace ozvučena, bylo třeba použít nepřenositelného proprietárního řešení) s sebou přineslo specifikování nového *profilu*, a to *MIDP 2.0* [9].

*MIDP 2.0* klade oproti *MIDP 1.0* větší nárok na paměť, a to navýšení o 128kB RAM a o 96kB více pro JRE běh [5].

Rozhraní *MIDP 2.0* rozšiřuje *MIDP 1.0* o následující [9]:

- HTTPS (tedy šifrovaný HTTP protokol), dále je doporučena (ne však striktně zaručeno) implementace socketů a datagramů (i v šifrované podobě).
- Podpora zvuku.

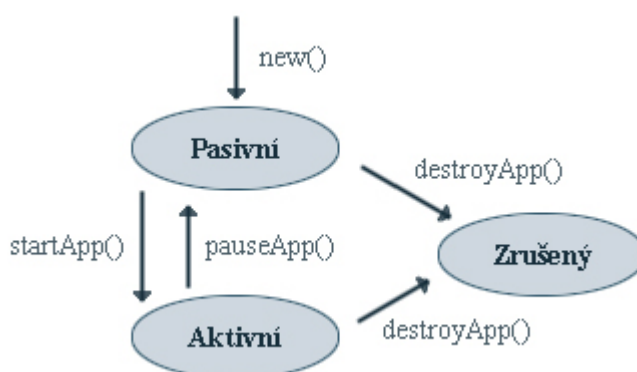
- UTF8 kódování.
- Možnost celoobrazového módu (*fullscreen mode*), třída `Canvas` má k dispozici celý `display`.
- Herní rozhraní. Jedná se pouze o zjednodušení práce. Tyto třídy by šly (pracně) naimplementovat v *MIDP 1.0*.
- Sdílení dat s MIDlety jiných sad.

Jak již bylo zmíněno, mobilní aplikace **Mariáš** používá z důvodu větší čitelnosti celoobrazové zobrazení a je tedy nutné použít profil *MIDP 2.0*, jenž je dnes dostupný na téměř všech mobilních telefonech z posledních let [7].

## 1.4 MIDlet

Aplikacím naprogramovaným v J2ME pro *MIDP* se často říká *MIDlet*. Je to způsobeno tím, že hlavní třída musí dědit abstraktní třídu `javax.microedition.midlet.MIDlet`. V rámci této třídy se nachází abstraktní rozhraní metod `startApp()`, `pauseApp()` a `destroyApp(boolean unconditional)`, které se musí naimplementovat v dědicí třídě. Tyto metody implikují stav právě spuštěné aplikace. Přechod mezi stavy zajišťuje **aplikační manažér**.

Po zavolání konstruktoru se aplikace nachází v *pasivním stavu*. V tomto stavu by neměla vlastnit či používat žádné sdílené zdroje. Pro přechod z *pasivního stavu* do *aktivního stavu* je zavolána metoda `startApp()`. Při volání metody `pauseApp()`, kterou se aplikace vrací do *pasivního stavu*, by měla aplikace tyto zdroje opět uvolnit. Při ukončení aplikace zavolá **aplikační manažer** metodu `destroyApp(boolean unconditional)`. [5]



Obrázek 1.2: Řízení běhu aplikace Aplikačním manažérem



## 2 Karetní hra Mariáš

Tato kapitola krátce pojednává o pravidlech karetní hry mariáš. Existuje více variant mariáše a každá sestává z několika dle principu odlišných her. Tyto varianty se hrají ve dvou až čtyřech hráčích. Aplikace **Mariáš** implementovala nejzákladnější, avšak nejnáročnější, typ hry varianty Volený mariáš, jenž je původní variantou této karetní hry. Dále popisovaná pravidla platí právě pro Volený mariáš.

### 2.1 Obecná pravidla karetní hry Mariáš

Mariáš je zdvihová karetní hra, která se hraje v 32 karetními listy, jenž jsou rozděleny do čtyř barev (srdce, listy, žaludy, kule) po osmi hodnotách (VII, VIII, IX, X, spodek, svršek, král a eso). Pro zajímavost – hra má bohatou bázi slangových výrazů a hlášek, jenž se postupem času infiltrovaly do běžné řeči a jejich používání se již zažilo i mezi neznalci mariáše (trumf, nechat ve štychu, přiznat barvu, atd.). Každý hráč je při každé hře označen speciálním názvem – *rozdávající* má po levé ruce *forhonta* a po pravé *zadáka*. V následujícím kole se pořadí posune ve směru hodinových ručiček. Karty se rozdávají tak, že forhontovi se nejprve dá 7 karet, dalším hráčům 5 a následně všem po pěti kartách.

Forhont začíná hru zvolením barvy (trumfu) jednou z prvních sedmi karet, případně z dalších pěti, jejichž hodnoty ovšem zatím nezná. Dále odloží dvě karty ze všech dvanácti do *talonu*, s nímž se dál nehraje, přitom nesmí odkládat desítky, esa ani trumfy. Spoluhráči následně souhlasí nebo nesouhlasí se zvolenou barvou.

V případě nesouhlasu hraje nesouhlasící hráč *betl* nebo *durch*. V tomto případě vezme daný hráč talon a zvolí nový (libovolný). Následně si vybere, jestli chce hrát *betl* (zavazuje se nevyhrát ani jeden zdvih) nebo *durch* (zavazuje se vyhrát všechny zdvihy) a začíná všechny zdvihy. Při těchto hrách se nevolí barva a desítka má obvyklou hodnotu (mezi devítkou a spodkem).

Pokud oba spoluhráči souhlasí s barvou, mají možnost *flekovat*, případně se závazky (jimiž slibují uhrát sto a více bodů, trumfovou sedmu, apod.), čímž zvyšují cenu hry, o níž se hraje. Při této hře hrají *rozdávající* a *zadák* proti forhontovi, přičemž si nesmí radit nebo ukazovat karty.

První zdvih začíná forhont a pokračuje se ve směru hodinových ručiček. Hráč začínající zdvih smí zahrát libovolnou kartou (vložená karta), další hráči se musí řídit následujícími pravidly:

- Hráč musí ctít barvu vložené karty s tím, že ji musí přebít, pokud může.
- Pokud spoluhráč po vložené kartě zahrál trumfem, musí hráč zahrát libovolnou kartou vložené barvy (nemusí přebíjet), případně trumfem (nemusí přebíjet).
- Pokud hráč nemá barvu vložené karty, musí zahrát trumfem (nemusí přebíjet).
- Pokud hráč nemá ani barvu vložené karty, ani trumf, zahraje libovolnou kartou.

Po zahrání všech hráčů vyhrává zdvih hráč, který zahrál nejvyšším trumfem, nebo následně, když se nehrálo trumfem, hráč, který zahrál nejvyšší barvou vložené karty. Nejhodnotnější kartou je eso, následně pak desítka a dále již obvykle – král, svršek, spodek, devítka, osmička, sedmička. Další zdvih začíná ten hráč, jenž vyhrál zdvih předchozí.

Po zahrání všech karet se sčítají body – forhontovi samostatně, rozdávajícímu a zadákovi společně. Hodnocené karty jsou eso a desítka. Oboje jsou za 10 bodů, v případě trumfové barvy za 20 bodů. Dalšími deseti body se taktéž hodnotí poslední zdvih. Celkem je tedy ve hře 110 bodů. Strana s více body vyhrává, pokud současně dodržela svých závazků, jinak prohrála. Současně může prohrát i druhá strana, pokud taktéž nedostála závazkům.

Mariáš se zpravidla hraje o peníze. Nevsází se však tolik, jako například ve hrách Blackjack nebo Poker. Výhry z jednotlivých her dosahují maximálně desítek korun, častěji však výrazně méně.

## 2.2 Typy her

Zvolení hry zvyšuje cenu hry – sázky, o níž se hraje. Při nesplnění závazku tak může prohrát i hráč, který uhrál více bodů a zdánlivě tak zvítězil.

- **Hra** – Účelem této hry je nahrát více bodů, než soupeř. Hodnocené karty jsou desítky, esa a poslední zdvih
- **Sedma** – Podstatou je vyhrát poslední zdvih svou trumfovou sedmou a současně uhrát Hru.
- **Stovka** – Hráč se zavazuje uhrát aspoň sto bodů.
- **Stosedm** – Hráč chce uhrát hry Stovku i Sedmu.
- **Betl** – Hráč se zavazuje neuhrát ani jeden zdvih. Tento hráč začíná každý zdvih. Není volen trumf a desítka má svou obvyklou hodnotu (mezi devítkou a spodkem).
- **Durch** – Hráč se zavazuje uhrát všechny zdvihy. Tento hráč začíná každý zdvih. Není volen trumf a desítka má svou obvyklou hodnotu (mezi devítkou a spodkem).

## 3 Expertní systémy

Umělá inteligence počítače byla v aplikaci **Mariáš** řešena pomocí expertního systému. V této kapitole se nachází obecné pojednání o expertních systémech a principy v nich používané.

### 3.1 Obecné vlastnosti Expertních systémů

Expertní systém je počítačový program simulující činnost experta, tedy činnost rozpoznávací a rozhodovací při řešení úloh se snahou docílit rozhodnutí kvalitně odpovídající rozhodnutí samotného experta. Jejich úkolem je řešit problémy, které nelze řešit pomocí tradičních algoritmických postupů, tj. řešit nenumerické a neurčité informace. Základní součástí expertního systému tvoří znalostní báze a řídicí mechanismus. Obě tyto části jsou od sebe odděleny.

#### 3.1.1 Znalostní báze

Znalostní báze je explicitně vyjádřená báze dat expertních systémů využívaných při řešení úloh. Tato báze je jediný zdroj znalostí, které jsou uspořádány podle dané organizace a měla by být schopná přijímat nové, případně odstraňovat chybné a nepotřebné znalosti.

Je důležité, aby znalostní báze splňovala dva požadavky. Prvním je *modularita*, jejíž podstatou je zachovat bázi přehlednou pro její budoucí modifikaci. Druhým požadavkem je *požadavek strukturalizace*, jehož funkcí je vytvořit hierarchie pojmů vyskytujících se v bázi znalostí a při jeho splnění je následně snadné vyhledávání znalostí a přesun od konkrétního pojmu k obecnému a naopak.

Vyjádření znalostí se dá provést dvojím způsobem [10]:

1. *Deklarativně* – vyjadřují známé fakty nebo představují dotazy na konkrétní údaj a jsou proto označovány jako *poznatky*
2. *Procedurálně* – vyjadřují jakým způsobem se má z jednoho poznatku odvodit nějaký závěr a jsou označovány jako *pravidla*

V různých expertních systémech se tyto požadavky a způsoby různě kombinují.

#### 3.1.2 Řídicí mechanismus

Řídicí, neboli též *inferenční* – *odvozovací*, mechanismus je část expertního systému, jejímž úkolem je sbírat potřebná data, využívat dostupné znalosti a z nich vyvozovat expertní závěry. Zpravidla se jedná o část programu, která prohledává stavový prostor vyjadřující reprezentující znalosti výčtem pravidel, grafem, stromem, či jinak podobně. Občas se využívá i jiných specifických, neobecných principů a technik řízení.

Řídicí mechanismus používá k odvozování údajů těchto inferenčních metod [10]:

- *Dedukce* – odvozování závěrů, které přímo plynou z předpokladů
- *Indukce* – odvozování obecnějších poznatků ze specifitějších poznatků
- *Generování a testování* – získávání poznatků metodou pokus-omyl
- *Abdukce* – zpětné usuzování ze známého pravdivého závěru směrem k jeho předpokladům
- *Absence* – pokud chybí patřičná konkrétní znalost, implicitně se předpokládá nějaká obecná znalost
- *Analogie* – získávání závěrů na základě podobnosti situace vzhledem k jiné známé situaci

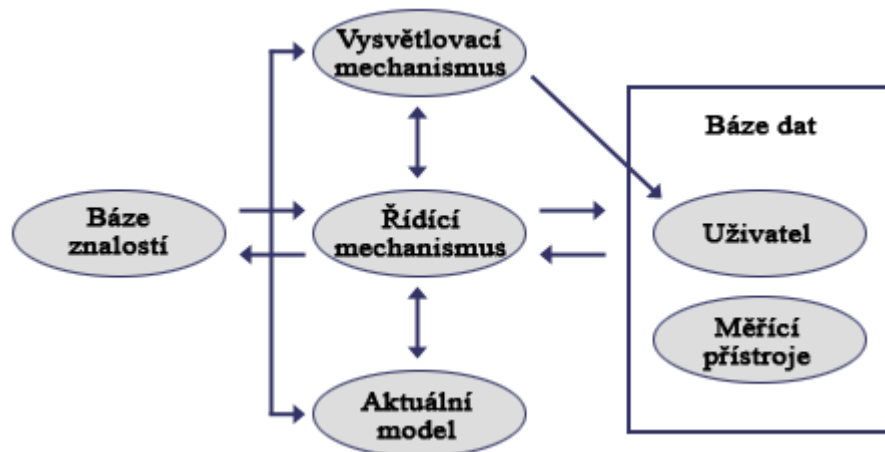
Podle typu řídicího mechanismu se expertní systémy dělí na *diagnostické (dopředné řetězení)* a *plánovací (zpětné řetězení)*. Jejich principy jsou vysvětleny v následující podkapitole. Dalším možným typem expertního systému je *hybridní expertní systém*, který kombinuje vlastnosti diagnostického a plánovacího. Tato kombinace však není sjednocením obou systémů, ale pouze o záměrné vyseparování vhodných vlastností.

## 3.2 Klasifikace expertních systémů na základě strategie procesu usuzování

### 3.2.1 Diagnostický expertní systém

Diagnostický expertní systém, neboli *dopředné řetězení*, určuje, které z možných řešení nejlépe odpovídá vstupním datům. V průběhu jeho činnosti se expertní systém dotazuje uživatele a z odpovědí vyvozuje dílčí závěry, z jejichž přehodnocování nakonec expert získává řešení problému.

*Báze dat* bývá složena ze zadaných dat od uživatele a také z naměřených hodnot. Vstupem konkrétních dat je model aktualizován. Na začátku jsou hodnoty vygenerovány expertem, po zadání dat uživatelem jsou tyto hodnoty v průběhu odvozování měněny. Funkcí *vysvětlovacího podsystému* je uchovávat a vysvětlovat postup, jímž bylo dosaženo výsledku. Jedná se tedy o oddělení uživatelského rozhraní od vlastního řídicího mechanismu. Řídicí mechanismus vybírá pouze ty dotazy, které mají největší váhu pro splnění dílčích hypotéz. Na *obrázku 3.1* je znázorněno blokové schéma popisovaného expertního systému.

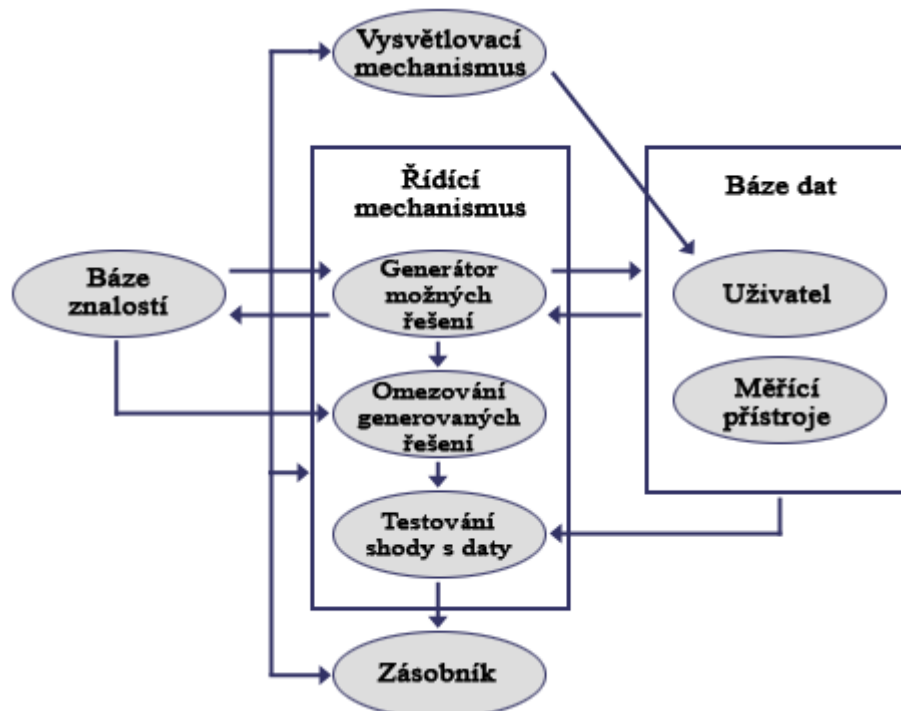


Obrázek 3.1: Blokové schéma diagnostického expertního systému

### 3.2.2 Plánovací expertní systém

Plánovací expertní systém, neboli *zpětné řetězení*, slouží k řešení problémů, u kterých známe předem požadovaný cíl. Jeho činností je tedy nalézt posloupnost kroků, s jejichž provedením lze cíle dosáhnout. Důležité je také zohlednit hlediska, pro něž by měla být hledaná posloupnost optimální.

Primární částí plánovacího expertního systému je *generátor možných řešení*, jenž generuje kombinace posloupností kroků, které jsou omezo­vány bází znalostí. Taktéž je testována shoda řešení s daty z báze dat. Výsledkem je množina možných postupů řešení, z nichž každému je přiřazen stupeň kvality. Na obrázku 3.2 je znázorněno blokové schéma tohoto expertního systému.



Obrázek 3.2: Blokové schéma plánovacího expertního systému

### 3.3 Expertní systém v aplikaci Mariáš

Umělá inteligence počítače při hře mariáše je řešena pomocí expertního systému. Jedná se o *diagnostický (dopředeně řetězený)* expertní systém se znalostní bází, jenž je uložena v *RMS* (Record Managment System) databázi. Vzhledem k povaze principů hry se dalo k implementaci řídicího mechanismu využít *deduktivní a induktivní inferenční metody*.

Expertní systém dostává v bázi dat informace o typu hry, zahraničích kartách spoluhráčů a vlastních kartách a na základě těchto údajů vyhodnocuje prohledáváním stavového prostoru, který je reprezentován stromem, nejvhodnější řešení, v tomto případě dle váhy sestupně uspořádanou množinu možných hratelných karet, z níž vybere právě tu nejvhodnější realizovatelnou.

Aplikace **Mariáš** je funkčně provedena tak, aby při případné nutnosti rozšířit znalostní báze stačilo uživateli aktivací příkazu v menu aplikace automaticky aktualizovat tuto bázi stáhnutím nových pravidel prostřednictvím HTTP komunikace z webového serveru, kde se aktualizace nachází (viz Kapitola 4.4 *Aktualizace UI z webového serveru*).

## 4 Midlet Mariáš

V rámci bakalářské práce byla vytvořena karetní hra **Mariáš** pro mobilní telefony. Následující kapitola se bude věnovat principům vývojem této aplikace. Jedná se tedy o implementační popis zabývající se realizací některých vlastností aplikace, zdůvodňováním realizace a možným eventuálně jiným řešením, nikoli o uživatelskou příručku popisující způsob užití – tento popis se nachází v příloze *Manuál použití aplikace* (viz *Příloha 1*).

### 4.1 Předběžné aspekty před tvorbou aplikace

Před samotným programováním je nutné, aby si vývojář uvědomil, pro jaké mobilní telefony a uživatele aplikaci vytváří. V kapitole *1.1 Úvod do J2ME* bylo zmíněno, že v edici J2ME se Java musí vyhnout snaze o jednotnou specifikaci z důvodů technických odlišností zařízení, třebaže se jinak snaží v ostatních edicích o interní sjednocení. Specifikace je tedy dána až určením dvojice *konfigurace a profilu*.

Bakalářská práce je zaměřena na mobilní telefony, pro něž je v drtivé většině určena konfigurace CLDC. Na úrovni konfigurace samotné existuje technické rozlišení na CLDC verze 1.0 a verze 1.1. Hlavním rozdílem mezi těmito verzemi je ten, že CLDC 1.0 nepodporuje plovoucí řadovou čárku, zatímco novější CLDC 1.1 již ano. Výběr správné konfigurace pro bakalářskou práci byl snadný, neboť počítání v plovoucí řadové čárce je pro **Mariáš** zbytečné. Navíc by se takové počítání dalo v jisté míře obejít pomocí celočíselného dělení a operace modulo. Realizace aplikace je tedy v souladu s konfigurací CLDC 1.0.

Důležitější rozhodnutí je až na úrovni profilu. Pro mobilní telefony se uvažují profily MIDP 1.0 a MIDP 2.0 (někdy též označované jako Java 1 a Java 2). Jelikož bylo z důvodu větší čitelnosti detailů na displeji rozhodnuto použít v aplikaci celoobrazové zobrazení, je nutné v aplikaci **Mariáš** použít profil MIDP 2.0 (viz Kapitola *1.3.2 MIDP 2.0*). Tímto krokem se znemožnilo přenášení na některé starší mobilní telefony, které nepodporují profil MIDP 2.0. Kdyby však bylo použito celoobrazového zobrazení z aplikace odstraněno, dalo by se využít profilu MIDP 1.0, čímž by se zaručila přenositelnost již na všechny mobilní telefony podporující J2ME a aplikace by byla tzv. *generická*.

Dalším neopomenutelným důležitým rozhodovacím prvkem je rozlišení displeje. Pro podporu spustitelnosti programu na větším počtu mobilních telefonů s různými rozlišeními displeje existují dvě základní metody tvorby aplikace, které se v praxi hojně uplatňují. Principem první metody je vytvoření více variant jedné aplikace, které se liší pouze rozlišením zobrazení. Obsahují tedy jen jednu obrázkovou sadu o daném rozlišení. Když si uživatel tuto aplikaci stahuje, musí vybrat pro sebe vhodnou verzi na základě typu svého mobilního telefonu, čímž je jednoznačně určena právě velikost

rozlišení displeje. Tyto aplikace jsou obecně graficky příjemnější, protože využívají co možno nejefektivněji daného rozlišení a taktéž mají menší velikost. Druhá metoda je pracnější, neboť se při ní vytvoří jen jedna aplikace, která se snaží být nezávislá na rozlišení uživatelského mobilního telefonu. Toho se docílí tak, že se použije více sad stejných obrázků, které se liší pouze v rozměrech a konkrétní obrázky se použijí až za běhu, kdy již aplikace zná rozlišení cílového zařízení, nebo se použije vestavěné vysokoúrovňové grafiky (jako například předdefinované formuláře) a grafických geometrických primitiv z nízkoúrovňové grafiky (obdélníky, úsečky, ...), kterým se udávají podle potřeby absolutní nebo relativní rozměry až za běhu. Občas bývá pro tvorbu grafického vzhledu použito kombinování obou předešlých technik. Nevýhoda použití výhradně jen druhého řešení spočívá v tom, že se kvůli použití grafických geometrických primitiv jedná o vzhledově méně nápaditou aplikaci nebo má aplikace kvůli použití několika obrázkových sad výrazně větší velikost. Výhodou však zůstává, že se jediná aplikace graficky přizpůsobí všem cílovým zařízením. Pro maximální možnou přenositelnost mezi mobilními telefony s různými rozlišeními displeje je aplikace **Mariáš** navržena za využití obrázků a vestavěné nízkoúrovňové grafiky (bližší popis docílení této techniky viz Kapitola 4.2 *Uživatelské rozhraní*).

Aplikace určená pro konfiguraci CLDC 1.0, profil MIDP 2.0 a obecné rozlišení displeje je takřka univerzálně přenositelná na veškeré mobilní telefony s podporou platformy J2ME. Tato míra přenositelnosti je však dána na základě nejobecnějšího charakteru. Taktéž je nutné uvažovat další vlastnosti, které budou probrány v dalších kapitolách.

Při vývoji aplikace je v neposlední řadě dobré inspirovat se u již existujících aplikací stejného nebo podobného typu, například u komerčních aplikací a rozšířit vlastní produkt o funkce, které konkurence nenabízí. Aplikace **Mariáš** je sice zřejmě unikátní svým zaměřením, ale již disponuje nestandardním rozšířením – aktualizace umělé inteligence importováním daných pravidel z webového serveru (viz Kapitola 4.4 *Aktualizace UI z webového serveru*).

## 4.2 Uživatelské rozhraní

Spuštěním MIDletu **Mariáš** je zavolána hlavní spouštěcí třída `MainMarias`, která dědí abstraktní třídu `MIDlet`, což je povinností jakékoliv mobilní aplikace (viz Kapitola 1.4 *MIDlet*). V této třídě jsou implementovány povinné metody `startApp()`, `pauseApp()` a `destroyApp()`.

Zavoláním metody `startApp()` dochází k zobrazení úvodního menu, jenž obsahuje tyto položky (popis implementace v následujících kapitolách):

- **Play** (generace nové hry **Mariáš**)
- **Settings** (nastavení vlastností hry)
- **Load saved game** (načtení uložené hry)
- **Help** (nápověda k aplikaci)
- **Credits** (informace o autorovi a samotné aplikaci)



- **Exit** (ukončení aplikace)

Položka Settings dále obsahuje:

- **Game mode** (změna protihráčů z PC na lidi a zpět)
- **Update AI** (aktualizace expertního systému z webového serveru)
- **Edit names** (editace jmen hráčů)

Grafická reprezentace menu v emulátoru J2ME Wireless Toolkit je na *obrázku 4.1*.



*Obrázek 4.1: Menu v emulátoru J2ME Wireless Toolkit*

Menu se v mobilních aplikacích obecně řeší jedním ze dvou způsobů – nízkourovňovou grafikou (obrázky, grafická geometrická primitiva) nebo vysokoúrovňovou grafikou (seznam položek, který zaručuje vyšší přenositelnost na mobilní telefony s různými rozlišeními displeje a jednodušší udržitelnost, protože seznam lze velmi lehce modifikovat). V aplikaci **Mariáš** je menu, ostatně jako zbytek aplikace, vyřešeno nízkourovňovou grafikou. Nevýhodou vysokoúrovňové grafiky, která reprezentuje jednotlivé položky seznamu jako pár popisku a obrázku, je totiž nejednotnost různých mobilních telefonů v zobrazovací velikosti obrázků těchto položek. Nízkourovňová grafika se s takovými problémy neseťkává, protože je čistě na programátorovi, jak který obrázek zobrazí.

Při tvorbě aplikací v J2ME (hlavně u nízkourovňové grafiky) se vestavěné fonty příliš nepoužívají, protože nelze přesně určit jejich velikost. Lze totiž využít pouze fontů s relativními rozměry – `SIZE_LARGE`, `SIZE_MEDIUM` a `SIZE_SMALL`, přičemž pouze platí `SIZE_LARGE > SIZE_MEDIUM > SIZE_SMALL` a nelze určit, jakou velikost tyto hodnoty ve skutečnosti představují. Konkrétní velikost vestavěných fontů je závislá na zařízení, na kterém aplikace běží. Z toho důvodu se dělají pro aplikace jejich vlastní sady fontů ve formě obrázků (viz *Obrázek 4.2*), jejichž vykreslení je dokonce rychlejší, než vykreslování samotných vestavěných fontů. Jedinou nevýhodou těchto speciálních fontů je jejich podíl na zvyšování celkové velikosti aplikace.



Obrázek 4.2: Vlastní sada fontů aplikace Mariáš

Jelikož se pro zobrazení používá nízkoúrovňová grafika (obrázky, grafická geometrická primitiva), jsou vstupy od uživatele reprezentovány jako nízkoúrovňové události.

K indikaci stisku konkrétní klávesy slouží metody pro oznámení události (stisk klávesy, přidržování stisknuté klávesy, uvolnění klávesy). Každé klávese pak odpovídá konkrétní klávesový kód, který představuje celé číslo. To umožňuje existenci jisté množiny vyjádření klávesových konstant pomocí J2ME, které odpovídají klávesám, které musí být obsaženy na libovolném mobilním telefonu (číslíce, \*, #, ...). Z důvodu přenositelnosti aplikace **Mariáš** je použito právě těchto vestavěných konstant. Bývá však problém řešení pohybu po aplikaci. Některé přístroje pro pohyb používají joystick, jiné speciální klávesy, apod. Na různých mobilních telefonech se tedy v důsledku technické nejednotnosti k docílení pohybu mapují 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.

## 4.3 Ukládání rozehrané hry

Pro možnost pozdějšího dohrání hry má uživatel k dispozici funkci k uložení aktuálního stavu rozehrané hry do perzistentních dat.

J2ME k ukládání dat nepoužívá souborový systém, ale RMS (Record Management System), což je vestavený systém správy záznamů (databáze), která se skládá z jednotlivých záznamů, které tvoří pole bajtů, jenž jsou v rámci jedné RMS jednoznačně indexovány číslem záznamu (ID – primární klíč). Implementace RMS zajistí, že všechny operace v databázi jsou atomické, serializované a synchronní. Při smazání aplikace dochází i k odstranění databáze, která byla touto operací vytvořena.

Rozehraný stav se do RMS databáze ukládá v následujícím tvaru: název záznamu, datum vytvoření záznamu, typ hry, trumfová karta, údaje o hráčích. Ačkoliv je pro záznam primárním

klíčem index ID, je pro uživatele přívětivější orientovat se v záznamech pomocí názvu, jenž si při ukládání sami zvolí. Tím se také název stává primárním klíčem, který musí být v databázi unikátní, což je ze strany aplikace hlídáno a zaručeno.

Vyvolání rozehrané hry tedy vybírá uživatel na základě jména stavu. Položka se následně z databáze smaže.

## 4.4 Aktualizace UI z webového serveru

Mobilní telefony podporující platformu J2ME musí taktéž podporovat protokol HTTP 1.1 pro komunikaci se sítí Internet, protože specifikace J2ME definuje nutnost implementace hlaviček GET, POST a HEAD tohoto protokolu. Takto tomu je z toho důvodu, že mobilní síť musí podporovat protokol HTTP 1.1 na aplikační síťové vrstvě, čímž je nezávislý na technické realizaci sítě, avšak nižší vrstvy jsou už z důvodu technické realizace sítě v plné kompetenci sítě mobilních telefonů a proto zde není jednoznačný přístup [5].

Z důvodu maximální přenositelnosti proto byla komunikace aplikace **Mariáš** implementována pomocí protokolu HTTP. Pro správnost komunikace je ovšem nutné mít na mobilním telefonu nastavený přístupový bod (Access Point) na Internet, nikoliv na WAP. V případě nastaveného přístupového bodu na WAP by bylo nutné mít k dispozici bránu, která by sloužila pro DNS k rozpoznávání URL.

Požadavek uživatele aktualizovat expertní systém naváže komunikaci s webovým serverem přes HTTP protokol. Nejprve se posílá požadavek na aktuální verzi na adresu [http://www.stud.fit.vutbr.cz/~xpump01/marias/getAI.php?version=ČÍSLO\\_VERZE](http://www.stud.fit.vutbr.cz/~xpump01/marias/getAI.php?version=ČÍSLO_VERZE). Komunikace s webovým serverem tedy probíhá pomocí GET hlaviček. Tento odesílací řetězec je interní záležitostí aplikace a uživatel se tak nemusí starat o správnost požadavku. Skript na straně serveru zjistí, zda-li je k dispozici novější verze a ve kladném případě odpoví mobilnímu telefonu zasláním aktuálních pravidel prostřednictvím proudu HTTP. Získaná data se následně převedou do RMS databáze, kde je umístěna znalostní báze expertního systému.

## 4.5 Settings, Help, Credits, Edit names

Položky Settings, Help, Credits a Edit names byly taktéž implementovány pomocí nízkoúrovňové grafiky. Nedochází tedy k žádnému dalšímu omezování přenositelnosti aplikace na mobilní telefony podporující CLDC 1.0, MIDP 2.0.

# 5 Webové systém pro bezplatnou distribuci mobilních aplikací

Kapitola zdůvodňuje účel tvorby webových stránek, které původně vznikly jako projekt do předmětu Informační systémy 2007/2008 a byly vytvořeny za pomoci jazyků HTML, CSS, PHP a MySQL.

## 5.1 Účel webových stránek

V síti internetu je dnes k nalezení de-facto vše. Tato věta platí bez výjimky i pro mobilní aplikace. Naneštěstí jsou tyto aplikace většinou komerční a občas bez povolení autora i nelegálně distribuované na různých osobních stránkách majitelů mobilních telefonů. Běžný uživatel tak často nemá možnost sehnat potřebnou aplikaci do svého mobilního telefonu, aniž by za ni musel „zbytečně“ platit nebo porušovat autorský zákon. A přitom existuje celá řada nadšených programátorů J2ME, kteří by se rádi o své výtvořky podělili s širší veřejností.

Právě z těchto důvodů by byla vhodná realizace systému, které by sdružovaly nekomerční mobilní aplikace, včetně jejich autorů a uživatelů.

## 5.2 Funkčnost webových stránek

Uživatel-vývojář má možnost si na webových stránkách založit vlastní účet, pod nímž může publikovat své vlastní aplikace s tím, že může rozhodovat o přístupových právech k dané aplikaci. Webový systém rozeznává hierarchii uživatelů – nepřihlášení, přihlášení, testéři a administrátor.

Posláním uživatelů testerů je testovat zveřejněné aplikace na vlastních mobilních telefonech, čímž mohou ověřit, zda-li nemá testovaná aplikace omezení v přenositelnosti, o nichž si autor sám není vědom. Tímto poznáním může autor aplikaci upravit dřív, než ji zveřejní široké veřejnosti a ušetřit si tak od špatné pověsti.

Z čehož plyne další možné uplatnění webu – neplacená sebereklama. Uživatel-vývojář publikováním aplikací o sobě dává vědět jak mezi běžnými lidmi, kteří navštíví webové stránky, tak mezi lidmi z oboru a vytvářet si vlastní referenční materiál. Shlukování případných nadaných programátorů v platformě J2ME může vést ke spuštění impulsivních nebo dlouho chystaných, doposud nerealizovaných, projektů, které mohou stanovit profesní kariéru programátorů a objevit tak skryté přednosti.

Neposlední funkcí webového systému je možnost komunikace uživatelů s vývojáři a dát tak vzniknout skutečně poptávané aplikaci.

## 6 Porovnání s cizími aplikacemi

Mobilní aplikace **Mariáš** vytvořená v rámci bakalářské práce byla porovnána s cizími aplikacemi. Vzhledem k tomu, že porovnávaný **Mariáš** vyvinutý v platformě J2ME je pravděpodobně jediná existující aplikace pro mobilní telefony s touto hrou, srovnání bylo prováděno pouze s jinými karetními hrami pro mobilní telefony (solitaire, poker, oko) a dvěma mariášovými programy určenými pro PC. Dvě z pěti mobilních aplikací byly komerční a obě PC aplikace byly nekomerční.

Menu komerčních karetních her je obdobné s menu **Mariáše**, avšak využívají více obrázkových prvků a zvukové efekty, což mimo jiné zvyšuje celkovou velikost aplikace. To je patrné zvláště u striptýzového pokeru. Jejich přenositelnost je používáním specifitějších prvků snižována. Jedna aplikace navíc dostatečně správně nezobrazovala celý obsah grafického výstupu na displej. Tento problém je vysvětlitelný tím, že aplikace byla vytvořena už jen pro konkrétní model mobilního telefonu.

U nekomerčních aplikací je menu děláno za použití vysokoúrovňové grafiky (předdefinované formuláře, textová pole, ...) a jejich obecná přenositelnost je tímto zaručena. Naopak použití nízkourovňové grafiky ve hře samotné jsou občas kvůli špatnému pozicování vykreslované obrázky mimo displej mobilních telefonů s nižším rozlišením. Jedna aplikace navíc nepodporovala ovládání joystickem.

Odporovaná umělá inteligence u nekomerčních aplikací zjevně trpěla nevyspělostí a počítač tak relativně často kvůli své vlastní chybě zbytečně prohrával. Dokonce se občas u jedné nekomerční aplikace stávalo, že při volení karty počítačem došlo k zacyklení programu a hra se tím stala nedohratelyou.

Oba mariášové programy určené pro PC byly z uživatelského hlediska zpracovány vhodně. S výjimkou u jednoho, u něž byla hra doprovázena poměrně nepříjemnými zvuky, jejichž zakázání nebylo možné. Umělá inteligence v jednom případě byla vyvinutá dostatečně kvalitně, aby počítač dokázal hrát důstojně. Ve druhém případě však počítač hrál zjevně náhodně, pouze s omezením na pravidla zahratelynosti karty. Největší předností těchto aplikací je velikost rozlišení, s níž disponuje PC a mnohonásobně převyšuje možnosti mobilních telefonů.

Z hlediska porovnání aplikace **Mariáš** s komerčními aplikacemi karetních her, dá se usoudit, že **Mariáš** je obecně funkčně na jejich úrovni. Všechny tyto komerční aplikace disponují uživatelsky příjemným prostředím a netriviální umělou inteligencí. Navíc umožňovaly uchovávání meziher v perzistentní paměti pro možné následné dohrání.

Nekomerční mobilní aplikace byly funkčně často velmi omezené na nutné minimum a i když by bylo u dvou možné uchovávat výsledky o nejlepších hrách, tato možnost nebyla přístupná. Z možností využití vysokoúrovňové grafiky, která umožňuje velkou přenositelnost, v celkovém důsledku nebylo dobře využito vytvářením omezování ve hře samotné.

## 7 Závěr

Cílem bakalářské práce bylo vytvořit aplikaci Mariáš pro mobilní zařízení s ohledem na dále snadně modifikovatelnou umělou inteligenci. Aplikace byla řešena takovým způsobem, aby byla zachována co největší možná přenositelnost na mobilní telefony s podporou platformy J2ME s jediným omezením – vzhledem k různorodosti rozlišení displejů zachovat co možno nejpříjemnější uživatelské rozhraní.

Umělá inteligence je řešená přes expertní systém, jehož znalostní báze je při prvním spuštění uložena v RMS databázi mobilního telefonu. Díky logické struktury expertních systémů je možné další rozšiřování znalostí počítače bez zásahu do zdrojového kódu samotné aplikace.

Za využití tohoto poznatku bylo možné aplikaci rozšířit o nestandardní funkci online aktualizace znalostní báze prostřednictvím HTTP komunikace s webovým serverem, který obsahuje distribuované aktualizace. Tímto způsobem je možné umělou inteligenci dále vylepšovat, aniž by uživatel potřeboval instalovat aplikaci znovu, čímž by mohl ztratit další informace v RMS.

V rámci bakalářské práce byly navíc vytvořeny webové stránky, jejichž účelem je sdružovat nekomerční mobilní aplikace, jejich autory a uživatele na jednom místě přehledným způsobem a tím zvětšit konkurenci komerčním mobilním aplikacím, které svým množstvím vedou a nutí uživatele k placení občas za nepoužitelné zbytečnosti.

Aplikace by se v budoucnu dala rozšířit o další typy her mariáše, které jsou implementačně, relativně k obsáhlé hře, méně náročné. Dalším možným rozšířením by bylo realizovat rozšíření typu hry složené z lidí pomocí BlueTooth komunikace, čímž by se zvýšila plynulost samotné hry. Tato možnost by ovšem snížila přenositelnost aplikace na poměrně velkou množinu mobilních telefonů. Ne všechny přístroje s podporou platformy J2ME současně podporují BlueTooth a navíc práce s touto technologií není v rámci všech výrobců mobilních telefonů sjednocená.

Aplikace již byla rozšířena mezi známými a ti ji s nemalým zájmem v praxi testují na svých rozličných mobilních telefonech. S výjimkou jednoho staršího modelu s podporou platformy J2ME, která nepodporovala MIDP 2.0 profil, fungovala aplikace Mariáš bez problému.

# 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] Sun Java ME Technology. Java Sun Technology [online]. 2008 [cit. 2008-05-02].  
Dostupné z WWW: <<http://java.sun.com/javame/technology/index.jsp>>
- [3] Sun Java Wireless Toolkit 2.5. Java Sun Technology [online]. 2007-01-29 [cit. 2008-05-02].  
Dostupné z WWW: <[http://java.sun.com/products/sjwtoolkit/download-2\\_5.html](http://java.sun.com/products/sjwtoolkit/download-2_5.html)>
- [4] Sun Java CLDC. Java Sun Technology [online]. 2005-02 [cit. 2008-05-02]. Dostupné z WWW:  
<[http://java.sun.com/j2me/docs/pdf/CLDC-HI\\_whitepaper-February\\_2005.pdf](http://java.sun.com/j2me/docs/pdf/CLDC-HI_whitepaper-February_2005.pdf)>
- [5] Bittnerová, L., R., Procházka, J., Kovařík, P. Vývoj aplikací v J2ME. Interval [online].  
2005-06-13 [cit. 2007-05-02]. Dostupné z WWW: <<http://interval.cz/vyvoj-aplikaci/j2me/>>
- [6] Java Community Process, JSR#36. Java Community Process [online].  
2008 [cit. 2007-05-02]. Dostupné z WWW: <<http://www.jcp.org/en/jsr/detail?id=36>>
- [7] The Java ME Device Table. Java Sun Technology [online]. [cit. 2008-05-02].  
Dostupné z WWW: <<http://developers.sun.com/techtopics/mobility/device/device>>
- [8] MIDP 1.0, JSR 37. Java Sun Technology [online]. 2004 [cit. 2008-05-02].  
Dostupné z WWW: <<http://jcp.org/aboutJava/communityprocess/final/jsr037/index.html>>
- [9] MIDP 2.0, JSR 118. Java Sun Technology [online]. 2004 [cit. 2008-05-02].  
Dostupné z WWW: <<http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html>>
- [10] Mařík, V., Štěpánková, O., Lažanský, J.: Umělá inteligence 2. Nakladatelství Academia, Praha  
1997. ISBN: 80-200-0504-8.

# Seznam příloh

Příloha 1. Manuál použití aplikace

Příloha 2. CD (zdrojové kódy, programová dokumentace, technická zpráva)



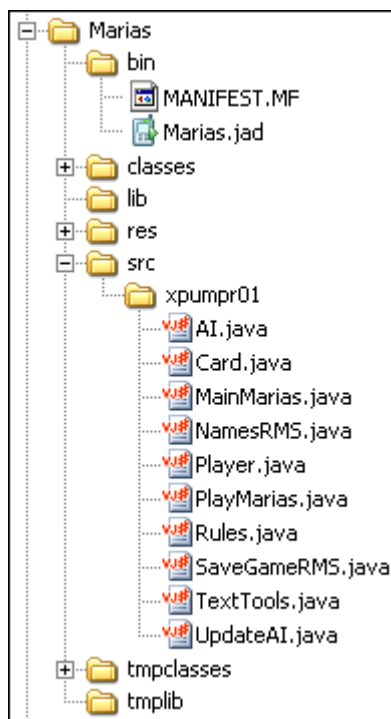
# Příloha 1 – Manuál použití aplikace

## Kompilace

Toto je pouze doporučený (osvědčený) postup: Za použití vývojového prostředí *J2ME Wireless Toolkit* je možné aplikaci nejen kompilovat, ale taktéž emulovat. Instalační balík tohoto nástroje je k dostání zdarma na webových stránkách Java Sun [3].

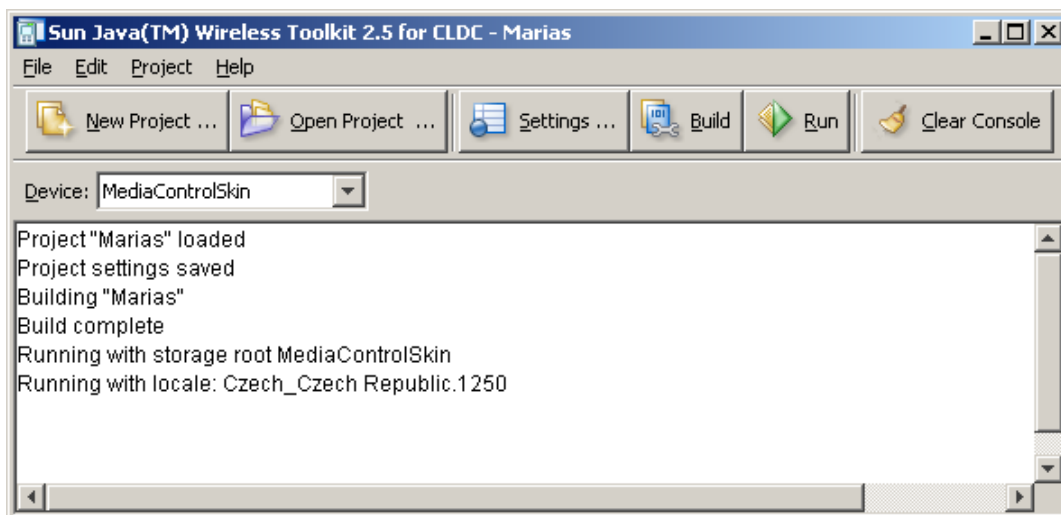
Po instalaci vývojového prostředí přepokopírujte adresář *Marias*, který je přiložen na médiu, do adresáře *apps*, který je součástí adresáře *WTK25*, jenž byl vytvořen po instalaci vývojového prostředí. V adresáři *apps* se nachází veškeré naprogramované aplikace určené pro kompilaci a spuštění.

Adresář *Marias* obsahuje vše potřebné pro úspěšnou kompilaci. Jedná se o zdrojové soubory (`./Marias/src/xpumpr01/*.java`), obrázky použité v aplikaci (`./Marias/res/*`), JAD soubor – Java Application Descriptor (`./Marias/bin/Marias.jad`) a soubor manifestu (`./Marias/bin/Manifest.mf`). Adresářová struktura je zobrazena na *obrázku A.1*.



Obrázek A.1: Adresářová struktura projektu *Marias*

Ve vývojovém prostředí *J2ME Wireless Toolkit* (Obrázek A.2) se projekt otevře tlačítkem *Open Project*. Zde se ze seznamu vybere projekt s názvem *Marias* (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 pomocí tlačítka *Build* (Obrázek A.2) zkompiluje projekt *Marias* (vytvoří se příslušné \*.class soubory). V tomto stavu je již aplikace ve vývojovém prostředí *J2ME Wireless Toolkit* emulovatelná.

Pokud se má aplikace nahrát do mobilního telefonu, je nutné vytvořit *jar* soubor (*Marias.jar*). Jedná se o zkomprimovaný soubor, který obsahuje přeložené \*.class soubory, obrázky, manifestový soubor a jiné podpůrné soubory, které jsou nutné k chodu aplikace. Jeho vytvoření se v *J2ME Wireless Toolkit* provádí následujícím postupem: *Project* → *Package* → *Create Package*. Nyní je *Marias.jar* vytvořen (`./Marias/bin/Marias.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 tlačítkem *Run* (Obrázek A.2).

## Menu

Jakmile se aplikace spustí, je zobrazeno hlavní menu (Obrázek 4.1).

V menu lze vybrat následující položky:

- **Play** (generace nové hry Mariáš)
- **Settings** (nastavení vlastností hry)
- **Load saved game** (načtení uložené hry)
- **Help** (nápověda k aplikaci)
- **Credits** (informace o autorovi a samotné aplikaci)
- **Exit** (ukončení aplikace)

# Play

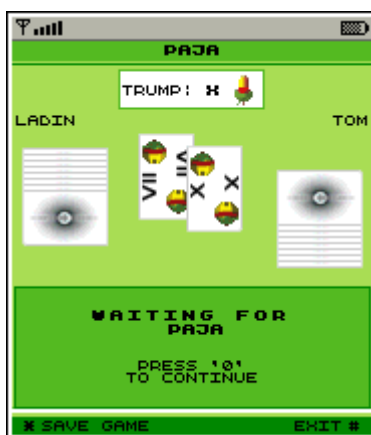
Při spuštění hry se v závislosti na zvoleném typu hry zobrazuje pohled hráče. Pokud se jedná o hru proti počítači, je po celou dobu pohled ze strany člověka. V případě, že hrají tři lidé, pohled se aktualizuje podle aktivního hráče a před každou zahrnou kartou jsou karty v hráčově ruce schovány před zraky spoluhráčů (viz *Obrázek A.4*). Zobrazování hrací plochy je tedy vždy z pohledu „*first person*“.

Po kartách se posunujeme šipkami / joystickem do stran nebo klávesami 4 a 6, označujeme šipkou / joystickem nahoru nebo klávesou 2 a odznačujeme šipkou / joystickem dolů nebo klávesou 8. Zvolenou kartu zahrajeme klávesou Enter / zmáčknutím joysticku nebo klávesou 5. Dále pak klávesou \* (hvězdička) můžeme rozehranou hru uložit nebo klávesou # („křížek“) hru ukončit.

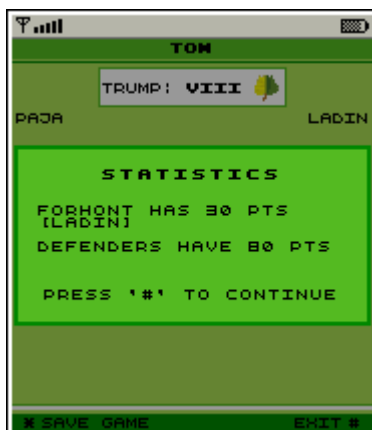
Při volení trumfu a talonu je forhont upozorněn zprávou (viz *Obrázek A.3*). Po zvolení trumfu se trumfová karta zobrazí nahoře displeje. Následující tahy jsou kontrolovány, zda-li jsou podle pravidel a tak je nemožné zahrát proti pravidlům. Po zahrání všech karet se zobrazí statistika se získanými body (viz *Obrázek A.5*).



*Obrázek A.3: Volba trumfu*



*Obrázek A.4: Hra tří lidí – čekání na hráče*



Obrázek A.5: Statistika dohrané hry

## Settings

*Settings* obsahuje možnosti dalšího nastavení aplikace.

V *Settings* lze vybrat následující položky (Obrázek A.6):

- **Game mode** (změna protihráčů z PC na lidi a zpět)
- **Update AI** (aktualizace expertního systému z webového serveru)
- **Edit names** (editace jmen hráčů)
- **Back** (zpět do menu)



Obrázek A.6: Grafická reprezentace nabídky Settings

*Game mode* umožňuje přepínat mezi tím, zda-li spoluhráči budou lidé nebo počítač.

*Update AI* po připojení na Internet automaticky zjišťuje, jestli existuje novější (lepší) verze počítačové umělé inteligence a při její existenci danou aktualizaci provede.

*Edit names* dává uživateli možnost měnit jména hráčů dle své vlastní libosti (viz Obrázek 4.2).

## Load saved game

Uživatel má možnost dohrát již v minulosti uloženou hru. Vybírá se ze seznamu názvů uložených stavů her (viz *Obrázek A.7*). Pokud se některý uložený stav vybere, dochází k jeho smazání a uživatel pokračuje v znovuustavené rozehrané hře tam, kde před tím skončil.



Obrázek A.7: Grafická reprezentace nabídky Load game

## Help

Obsahuje popis pravidel karetní hry Mariáš a její ovládání.

## Credits

Obsahuje informace o autorovi a kontakt na něj a základní informace o aplikaci (viz *Obrázek A.8*).



Obrázek A.8: Grafická reprezentace položky Credits

## Exit

Ukončuje aplikaci.