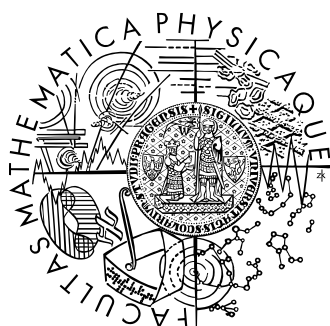


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Marek Štalmašek

**MobilBomber**

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Michal Malohalva

Studijní program: Informatika, Obecná Informatika

2010

V prvom rade by som chcel poďakovať svojmu vedúcemu RNDr. Michal Malohlava, za rady a poznámky, ktoré mi poskytol pri písaní Bakalárskej práce a s ňou spojeným programom. Taktiež by som chcel poďakovať rodičom a celej rodine za podporu a nápady, ktoré mi poskytovali po celú dobu písania práce.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa

Marek Štalmašek

# Obsah

|   |    |
|---|----|
| 1 Úvod.....   | 6  |
| 1.1 Motivácia.....  | 6  |
| 1.2 Ciele práce.....  | 7  |
| 1.3 Štruktúra práce .....                                     | 8  |
| 2 Technológie.....  | 10 |
| 2.1 Technológia Bluetooth .....                               | 10 |
| 2.2 Technológia WiFi.....                                     | 15 |
| 2.3 Technológia IrDA .....                                    | 17 |
| 2.4 Porovnanie bezdrôtových technológií.....                  | 19 |
| 2.5 J2ME a obmedzenia mobilných telefónov .....               | 20 |
| 3 Analýza problému .....                                      | 23 |
| 3.1 Popis problému .....                                      | 23 |
| 3.2 Komunikačné technológie podporované hrou MobilBomber..... | 27 |
| 4 Návrh riešenia .....  | 32 |
| 4.1 Rozhranie .....   | 32 |
| 4.2 Komunikačný modul.....                                    | 33 |
| 4.3 Hra MobilBomber .....                                     | 33 |
| 5 Užívateľská dokumentácia.....                               | 35 |
| 5.1 Inštalácia programu.....                                  | 35 |
| 5.2 Ovládanie a pravidlá hry .....                            | 36 |
| 6 Programátorská dokumentácia.....                            | 41 |
| 6.1 Štruktúra programu .....                                  | 41 |

|   |    |
|---|----|
| 6.2 WirelessConnectionLibrary .....     | 41 |
| 6.3 MobilBomber .....                   | 44 |
| 6.4 Preklad a zostavenie programu ..... | 47 |
| 6.5 Prepojenie pomocou WiFi .....       | 48 |
| 6.6 Prepojenie pomocou Bluetooth .....  | 51 |
| 6.7 Testovanie programu.....            | 56 |
| 7 Známe sieťové hry .....               | 58 |
| 7.1 Snake II .....                      | 58 |
| 7.2 Tennis Smash .....                  | 59 |
| 7.3 Hry podporujúce WiFi .....          | 59 |
| 8 Záver .....                           | 60 |
| Literatúra .....                        | 62 |
| Príloha A .....                         | 64 |
| Adresárová štruktúra CD.....            | 64 |
| Príloha B .....                         | 66 |
| Testované telefóny .....                | 66 |

Názov práce: MobilBomber

Autor: Marek Štalmašek

Katedra (ústav): Katedra softwarového inženýrství

Vedúci bakalárskej práce: RNDr. Michal Malohlava

e-mail vedúceho: [michal.malohlava@dsrg.mff.cuni.cz](mailto:michal.malohlava@dsrg.mff.cuni.cz)

Abstrakt: Video hry sú dnes nedeľnou súčasťou mobilných telefónov. Špeciálne odvetvie tvoria sieťové hry, ktoré sa dajú rozdeliť na krátko a ďaleko dosahové, podľa použitých komunikačných technológií. Táto práca sa zameriava na bezdrôtové krátkodosahové siete, akými sú Bluetooth, WiFi, IrDA. Jej cieľom je vytvoriť v jazyku J2ME real-time sieťovú mutáciu hry typu Bomberman, postavenú na základe univerzálneho rozhrania tak, že hra samotná nebude závislá od konkrétnej bezdrôtovej technológie. Univerzálne komunikačné rozhranie bude implementované v samostatnej knižnici pre každú technológiu. To si vyžaduje dôkladné pochopenie bezdrôtových technológií, ktorému sa tiež práca venuje.

Názov práce: MobilBomber

Author: Marek Štalmašek

Department Katedra softwarového inženýrství

Supervisor: RNDr. Michal Malohlava

Supervisor's e-mail: [michal.malohlava@dsrg.mff.cuni.cz](mailto:michal.malohlava@dsrg.mff.cuni.cz)

Abstract: Nowadays, video games are a big part of mobile phone software industry. Special place holds multiplayer video games, which can be divided to short-range and long-range. This work aims to short-range wireless network technologies as Bluetooth, WiFi and Infrared. The main goal of the work is to create a real-time multiplayer variation of Bomberman game based on universal network interface in that way, so it will be totally independent on particular wireless network technology. Universal communication interface, will be implemented by a separate module for each network technology. Therefore a deep understanding of short-range wireless communication is required, which the focus of this work is also targeted on.

# Kapitola 1

## Úvod

### 1.1 Motivácia

Prvé mobilné telefóny boli svojim primárnym účelom určené na bezdrôtovú komunikáciu. Uplatnenie si našli v nepriaznivom teréne a miestach bez pokrytia klasickou telefónnou linkou. Medzi ľuďmi, ktorí sú neustále v pohybe a komunikácia je dôležitou súčasťou ich profesionálneho života. Od tej doby ubehlo veľa rokov.

Súčasný trendy naznačujú že mobilné telefóny sa stanú multifunkčnými zariadeniami, bez ktorých si človek nebude vedieť predstaviť svoj život [1]. Už teraz sa mobilné telefóny približujú mobilným PDA. Fotoaparát, ktorý svojím rozlíšením a kvalitou pomaly dobieha kompaktné digitálne fotoaparáty a MP3 prehrávač aj s podporou iných hudobných formátov, sa stáva štandardom. Súčasťou mobilných telefónov sú aj kalendáre, kalkulačky, diktafóny a iné vymoženosti, ktoré ľuďom dokážu významne šetriť čas.

Taktiež po technickej stránke zaznamenali mobilné telefóny významný krok vpred. Už to nie sú kilogram vážiace zariadenia, ktoré majú neestetický vzhľad a jedno- až dvoj- riadkový čiernobiely displej, zobrazujúci len znakovú grafiku. Displeje sa zmenili na plne grafické farebné rozhrania s relatívne vysokým rozlíšením (320 x 240 pixlov) a veľkou farebnou hĺbkou (aj 24bitovou), na ktorých je možné vierohodne zobrazovať videá a fotografie. Zlepšila sa kvalita reprodukovaneho zvuku a pribudli ďalšie komunikačné rozhrania ako IrDA a Bluetooth, prípadne v súčasnosti si svoju cestu raziace WiFi.

Tento vývoj otvoril dvere dokorán pre herný priemysel, ktorý podľa viacerých štúdií v blízkej budúcnosti predbehne filmový priemysel [2]. Mobilné telefóny sú práve tým smerom, ktorý by tomu mohol napomôcť.

V súčasnosti vzniká mnoho hier, ktoré sa síce pýšia slávnymi menami, ale nedosahujú kvalitu svojich predchodcov zo sveta herných konzol a počítačov. Často sú to nepodarené reedície so slabou hrateľnosťou a niekedy až nepochopiteľným ovládaním na malých klávesniciach mobilných telefónov.

Lepšiu hrateľnosť a zábavu do mobilných hier by mohol priniesť multiplayer mód, teda možnosť hry viacerých hráčov proti sebe. To je technologicky splniteľné práve vďaka už spomínaným novým komunikačným rozhraniam, ako je IrDA, Bluetooth a Wifi. Implementáciou takejto hry a porovnaním protokolov, týchto troch rozhraní, sa bude zaoberať práve táto práca.

## 1.2 Ciele práce

Hlavným cieľom práce je naprogramovať sieťovú aplikáciu real-time multiplayer hry typu Bomberman postavenej na krátkodosahových bezdrôtových sieťach. Konkrétne pre technológie Bluetooth, WiFi a Infračervené spojenie (IrDA). Aplikácia by sa skladala z komunikačnej časti (knižnice) a hry samotnej. Implementácia komunikačného modulu, podmieňujúca dostatočné pochopenie vyššie menovaných technológií, by mala byť nezávislá na časti implementujúcej samotnú hru. Na druhej strane hra by mala byť schopná pracovať so sieťovým rozhraním bez ohľadu na implementáciu konkrétnej technológie.

Čiastkové ciele teda sú:

- (C1) Navrhnuť komunikačné riešenie typu klient-server, ktoré by bolo kompatibilné s hore menovanými technológiami.
- (C2) Na základe tohto riešenia navrhnuť interface na programovej úrovni.
- (C3) Zanalyzovať bezdrôtové technológie z hľadiska využiteľnosti pri tvorbe real-time multiplayer hier pre mobilné zariadenia.

- (C4) Implementovať všetky pre tento účel použiteľné bezdrôtové technológie.
- (C5) Navrhnuť dizajn komunikácie inštancií hry a viacvláknový beh hry, tak aby aplikácia spĺňala požiadavky real-time multiplayer hry pre 2-4 hráčov.
- (C6) Implementovať podľa tohto návrhu hru, využívajúc komunikačné rozhranie, tak aby jej beh nebol závislý na konkrétnej bezdrôtovej technológii.

### 1.3 Štruktúra práce

Kapitola 2 a zaoberá podrobným prehľadom komunikačných technológií IrDA, Bluetooth a WiFi, ako aj ich stručnou históriou. Nachádza sa v nej aj vzájomné porovnanie technológií a prehľadná tabuľka vlastností a parametrov. Na konci kapitoly sa nachádza popis jazyka Java 2 Micro Edition, profil MIDP 2.0 určeného pre mobilné telefóny a komunikačné rozhranie jazyka Java pre prácu so skúmanými komunikačnými technológiami.

Kapitola 3 opisuje podrobnejšie skúmaný problém z programátorského hľadiska, ako aj z hľadiska technológií bezdrôtového spojenia. Rozoberá vybrané známe multiplayer hry pre mobilné telefóny.

Kapitola 4 predkladá čitateľovi návrh riešenia problému.

Kapitola 5 je užívateľskou dokumentáciou. Po jej prečítaní by mal užívateľ byť schopný nainštalovať J2ME aplikáciu do mobilného telefónu, spustiť ju a nastaviť sieťové spojenie. Pomocou nej sa taktiež naučí ovládať postavy v hre. Súčasťou tejto kapitoly sú aj pravidlá hry a typy a triky, ktoré môžu užívateľovi pomôcť poraziť súpera.

Kapitola 6 rozoberá mobilnú aplikáciu z programátorského hľadiska. Obsahuje použité dátové štruktúry, zaujímavé riešenia niektorých problémov, spojených s programovaním aplikácií pre mobilné zariadenia. Popisuje použité algoritmy dôležité pre chod hry a popis spojenia mobilných telefónov cez Bluetooth a WiFi.



Kapitola 7 popisuje známe implementácie týkajúce sa sieťových hier pre mobilné zariadenia využívajúce platformu JME. Spomenuté sú hry pre BT a IrDA. Pre absenciu WiFi implementácie podáva kapitola v závere vysvetlenie.

V Kapitola 8 obsahuje súhrn výsledkov mojich pozorovaní. Predpovedá možný vývoj hier pre mobilné telefóny a hodnotí vhodnosť jednotlivých komunikačných technológií na vytváranie hier pre viac hráčov. Posledná časť sa venuje kritickému zhodnoteniu nedokončených častí práce, ako aj možnému rozšíreniu práce.

# Kapitola 2

## Technológie

### 2.1 Technológia Bluetooth

Bluetooth [3] je otvorená štandardizovaná bezdrôtová technológia určená pre spájanie zariadení na krátke vzdialenosti (10m), ktorej pôvodnou úlohou bolo odbremeniť elektronické zariadenia, ktoré boli medzi sebou prepojené káblami. Mala to byť náhrada pre štandard RS-232 [22].

Po objavení všetkých jeho možností začala jeho obľuba rapídne rásť, pričom dnes je napríklad nedeliteľnou súčasťou každého lepšie vybaveného mobilného telefónu.

#### 2.1.1 Vlastnosti technológie

Bezdrôtová technológia Bluetooth sa vyznačuje niekoľkými špecifickými vlastnosťami.

Nasledujúce body zhŕňajú tie najdôležitejšie z nich.

- Krátkodosahový rádiový prenos – zariadenia spolu komunikujú po rádiových vlnách na frekvencii 2,4 GHz.
- Maximálna vzdialenosť jednotlivých zariadení môže byť až 100 metrov, spravidla býva ale iba 10 metrov.

- Nízky vysielací výkon (záleží na dosahu), takže vysielateľ v malom prenosnom zariadení s obmedzenou kapacitou akumulátora nespotrebovávajú veľa energie.

- Podpora nielen dátových, ale tiež hlasových prenosov (využitie napríklad v profile Headset pri spojení bezdrôtovej handsfree súpravy s mobilným telefónom).

V tabuľke 1 sa nachádzajú jednotlivé triedy Bluetooth a ich maximálny možný dosah. Ten sa môže líšiť podľa prostredia, kde prenos cez Bluetooth prebieha (bytovky, voľné priestranstvá, atď.). Dosah spojenia je závislý výkone Bluetooth zariadenia. Väčšina mobilných telefónov sa nachádza v triede Class 2, takže dosah ich Bluetooth zariadení činí zhruba desať metrov.

| Trieda  | Maximálny povolený výkon |        | Dosah (približný) |
|---------|--------------------------|--------|-------------------|
|         | (mW)                     | (dBm)  |                   |
| Class 1 | 100 mW                   | 20 dBm | ~ 100 m           |
| Class 2 | 2,5 mW                   | 4 dBm  | ~ 10 m            |
| Class 3 | 1 mW                     | 0 dBm  | ~ 1 m             |

Tabuľka 3: Dosah technológie Bluetooth prevzaté z [12]

### 2.1.2 História

Už v roku 1994 sa spoločnosť Ericsson začala zaoberať výskumom technológie, ktorá by umožnila bezdrôtové prepojenie mobilných zariadení ich príslušenstva. Hlavnými požiadavkami bola nízka cena komponentov, malý príkon a rozmery zariadenia. Onedlho sa k Ericsson pripojuje Intel, IBM, Nokia a Toshiba a v roku 1998 spolu vytvárajú skupinu zaoberajúcu sa výskumom bezdrôtovej technológie Bluetooth (Bluetooth Special Interest Group – SIG [5]), ktorá má za úlohu podrobne špecifikovať štandard Bluetooth. V júli 1999 Bluetooth SIG publikuje Bluetooth špecifikáciu verzie 1.0.

V dnešnej dobe je aktuálna špecifikácia v1.2 a už sa pracuje na v2.0 [6]. Bluetooth Special Interest Group má okolo 2 000 členov. Všetky informácie o technológii Bluetooth sú dostupné na internetových stránkach [7].

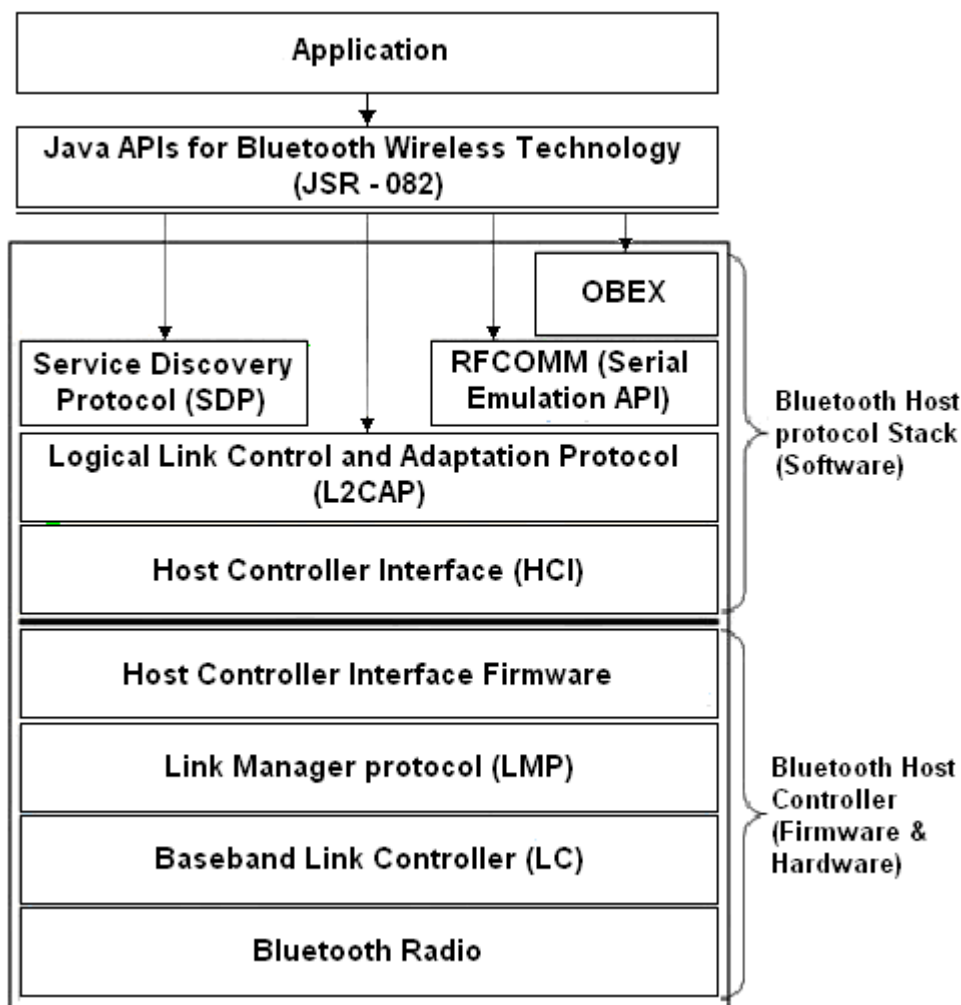
### 2.1.3 Protokol Bluetooth

Jadrom špecifikácie Bluetooth je tzv. „Bluetooth protocol stack“, ktorý definuje, ako celá technológia funguje. Rozdeľuje Bluetooth na jednotlivé vrstvy, podobne ako napríklad ISO model OSI u počítačových sietí. Jeho schému zobrazuje Obrázok 1, pričom nižšie vrstvy od tých vyšších oddeľuje tzv. HCI (Host Controller Interface). Najnižšie sa nachádza Radio vrstva, ktorá sa stará o moduláciu a demoduláciu signálu a popisuje fyzické požiadavky na Bluetooth vysielateľ a prijímač konkrétneho zariadenia. Baseband / Link controller vrstva sa stará o formátovanie dát do podoby vhodnej pre prenos vzduchom a synchronizáciu spojenia. Link manager vrstva nadväzuje a udržiava spojenie medzi zariadeniami. Obdobne ako v oblasti počítačových sietí, aj tu rozlišujeme komunikáciu podľa spôsobu naviazania.

#### **Existujú dva druhy komunikácie:**

Synchrónna komunikácia so spojovaním - Synchronous, Connection-Oriented (SCO), slúžiaca hlavne pre hlasovú komunikáciu (napríklad profil headset).

Asynchrónna komunikácia bez spojovania - Asynchronous, Connectionless (ACL), vhodná hlavne pre dátovú komunikáciu.



Obrázok 1: Bluetooth protocol stack prevzaté z [8]

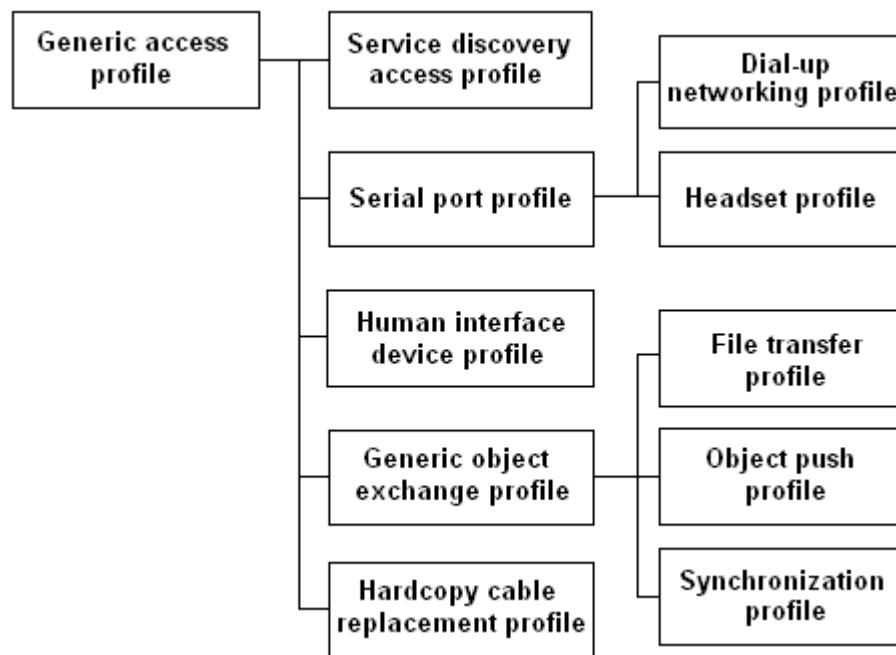
HCI (Host Controller Interface) sa stará o spoluprácu vyšších vrstiev s nižšími vrstvami (Radio, Baseband, Link manager). Medzi vyššie vrstvy patrí napríklad L2CAP (Logical Link Control and Adaptation Protocol), ktorá má na starosti zapuzdrenie paketov do formátu vhodného pre nižšie vrstvy, multiplexovanie spojenia tak, aby mohlo byť využívané viacerými aplikáciami, atď. SDP (Service Discovery Protocol) formuluje akcia pri ponúkaní a vyhľadávaní služieb Bluetooth zariadení. RFCOMM vrstva umožňuje emulovať prepojenie sériovým káblom, pričom má všetky špecifiká štandardu RS-232. Vďaka tomu môžu cez Bluetooth komunikovať aplikácie navrhnuté pre komunikáciu sériovým portom.

## 2.1.4 Bluetooth profily

Aby zariadenie mohlo používať technológiu Bluetooth, musí vedieť interpretovať jednotlivé Bluetooth profily. Tie sú tu od toho, aby bola eliminovaná prípadná nekompatibilita vyvíjaných programov u Bluetooth zariadení rôznych výrobcov. Každý profil zastupuje inú aplikáciu, inú úlohu resp. možné použitie technológie Bluetooth. Jeho špecifikácia musí zahŕňať:

- Závislosti na iných profiloch
- Navrhované formáty užívateľského rozhrania
- Časti Bluetooth protocol stacku využívané týmto profilom

Charakteristika niektorých profilov [9] je uvedená v nasledujúcom prehľade, pričom ich hierarchické usporiadanie zobrazuje Obrázok 2.



Obrázok 2: Hierarchia základných Bluetooth profilov prevzaté z [9]

Hlavným (base) profilom je GAP (Generic Acces Profile). Z neho vychádzajú všetky ďalšie profily, čím je zaručená všeobecná kompatibilita. GAP

zaisťuje mimo iného napríklad vyhľadávanie či naviazanie spojenia dvoch Bluetooth zariadení.

SDAP (Service Discovery Application Profile) popisuje, ako má aplikácia využiť SDP (viz Bluetooth stack protokol) pre vyhľadanie služieb na vzdialenom zariadení.

SPP (Serial Port Profile) definuje emuláciu RS-232 sériového rozhrania na Bluetooth zariadení.

DUN (Dial-Up Networking) je postavený na profile sériového portu a popisuje, ako má zariadenie využívať modem pre pripojenie k telefónnej sieti.

HSP (Headset Profile) popisuje spôsob komunikácie Bluetooth headset sady s počítačom či mobilným telefónom. Headset po prepojení funguje ako vstupne/výstupné vzdialené pripojené zariadenie.

GOEP (Generic Object Exchange Profile) – je základom pre ďalšie profily zaoberajúce sa výmenou dát, definuje role serveru a klienta.

## 2.2 Technológia WiFi

Technológia WiFi (čo je skratkou pre wireless fidelity) vznikla na prepojenie počítačov na krátku vzdialenosť. Pôvodne bolo zamýšľané, že nahradí ethernetový kábel v administratívnych budovách a bude náhradou pre „drôtovú“ LAN. Postupne sa prestala akcentovať väzba s ethernetom a začína sa všeobecne hovoriť ako o jednej variante sietí WLAN.

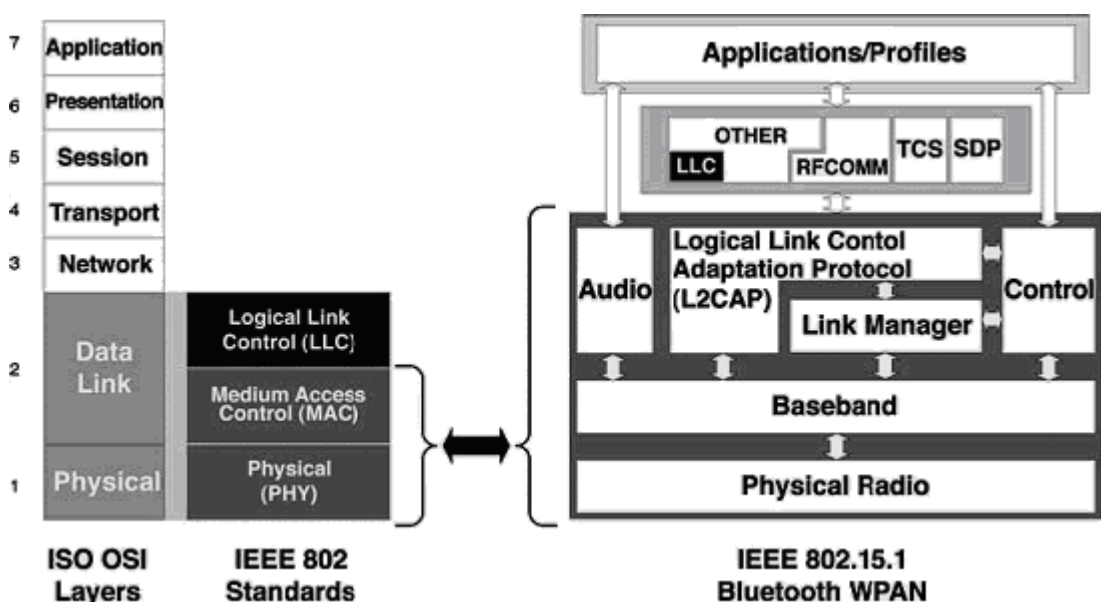
V súčasnosti je WiFi komplexné riešenie prepojenia viac zariadení bez použitia kabeľáže. Pomocou WiFi sú prepojené počítače na malú a strednú vzdialenosť – to znamená v rádoch metrov až desiatok metrov. Do WLAN sietí sa už dajú pripojiť tlačiarne, notebooky a PDA. Na trhu sa dokonca objavujú prvé mobilné telefóny podporujúce túto technológiu bezdrôtovej komunikácie.

## 2.2.1 História WiFi

V rokoch 1986 až 1987 sa objavujú prvé proprietárne riešenia napr.: Proxim, Symbol [10][11]. Využívajúc pásma ISM a UNII, sú navzájom nekompatibilné. Objavuje sa teda potreba spoločného štandardu, aby si proprietárne riešenia rozumeli navzájom.

V roku 1988 vzniká pracovná skupina IEEE 802.11, z iniciatívy spoločnosti NCR. Táto spoločnosť totiž chcela bezdrôtovo prepojiť svoje pokladne. V rokoch 1989-97 prebieha úporné hľadanie technického riešenia. V roku 1990 vzniká AT&T WaveLAN, ktorý na šírenie svojho signálu používa techniku DSSS (Direct Sequence spread spektrum). V roku 1996 je vyvinutý prvý chipset pre bezdrôtový Ethernet. PRISM WLAN chipset od Harris (Intersil).

V roku 1997 je dosiahnutá dohoda na spoločnom štandarde – IEEE 802.11 [12]. Tento štandard bol ale už pri zadávaní zastaraný a preto hneď začínajú práce na jeho vylepšení. V roku 1999 sú schválené nové štandardy. 802.11a má prenosovú rýchlosť 54 Mbit/s v pásme 5 GHz, ktoré je použiteľné v USA. Štandard 802.11b má prenosovú rýchlosť 11 Mbit/s a pracuje na frekvenčnom pásme 2,4 GHz.



Obrázok 3: Protokol WiFi a Bluetooth prevzaté z [12]



## 2.2.2 Protokol WiFi

WiFi je založená na takzvaných basic service set (ďalej BSS) bunkách, ktoré ju vytvárajú [12]. Tieto bunky môžu byť prenosné alebo pevné stanice. Prístup k prenosu zabezpečujú koordinačné funkcie a to menovite distribuovaná koordinačná funkcia (DCF) alebo bodová koordinačná funkcia (PCF) [15].

Šírka pásma je rozdelená do 14 prekrývajúcich sa pásiem, kde každé má 22 MHz. Iba 11 z týchto pásiem je dostupných v Spojených štátoch a 13 v Európe. Všetky zariadenia patriace do jednej BSS používajú jeden kanál. Na rozdiel od Bluetooth, ktoré používa FHSS, WiFi využíva na multiplexovanie direct sequence spread spektrum (DSSS), complementary code key (CCK) alebo orthogonal frequency – division multiplexing (OFDM).

Na obrázku 3 sa nachádza porovnanie protokolov WiFi podľa štandardu IEEE 802.11x a Bluetooth štandard IEEE 802.15.1.

## 2.3 Technológia IrDA

Infrared Data Association (ďalej len IrDA) špecifikuje 3 infračervené komunikačné štandardy:

- IrDA data
- IrDA control
- AIr

Pre prenos dát sa používa štandard IrDA – Data. Tento štandard sa používa aj na mobilných telefónoch a prístupom k prenosu dát je veľmi podobný technológii Bluetooth. Tak ako Bluetooth aj IrDA podporuje na vyššej aplikačnej vrstve výmenu objektov (tzv. OBEX protokol).

Ako bolo uvedené v tabuľke číslo 4, IrDA je síce technológia, ktorá umožňuje posielat' až 16 megabytov za sekundu, ale jej dosah je veľmi krátky. Zariadenia

musia mať priamu viditeľnosť, teda medzi nimi nesmie byť žiadna prekážka. Vzdialenosť jeden meter teda umožňuje prenos dát medzi zariadeniami. Neumožňuje ale aktívne používanie, akým je napríklad pripojenie handsfree k mobilnému telefónu alebo spojenie dvoch herných konzolí za účelom sieťovej hry pre viac hráčov.

## 2.4 Porovnanie bezdrôtových technológií

Bluetooth, WiFi a IrDA sú najbežnejšie bezdrôtové technológií, s ktorými sa dnes môžeme stretnúť. Infraport (IrDA) sa používa na prepojenie dvoch zariadení (dva mobilné telefóny navzájom, PC s mobilným telefónom, PDA s PC, atď.) na krátku vzdialenosť a priamu viditeľnosť. Rýchlosť tejto technológie je nízka a vyžaduje priamu viditeľnosť. Wireless LAN sa používa hlavne na pripojenie počítačov, resp. notebookov do lokálnej siete. Má pomerne veľký dosah (rádovo desiatky až stovky metrov), vysoké prenosové rýchlosti a nevyžaduje priamu viditeľnosť medzi zariadeniami.

Bluetooth, má dosah niekoľko (približne 10) metrov, nízky vysielač výkon a s tým spojenú nízku spotrebu napájacieho prúdu. Táto technológia sa pri sľúbenom zvýšení výkonu javí, ako perspektívna v veľa oblastiach elektrotechniky a mobilných zariadení.

| Technológia a funkcia  | Infrared                 | 802.11             | HomeRF             | Bluetooth™         |
|------------------------|--------------------------|--------------------|--------------------|--------------------|
| Typ spojenia           | Infračervené, priamy lúč | Rozšírené spektrum | Rozšírené spektrum | Rozšírené spektrum |
| Spektrum               | Optické 850 nm           | RF 2,4GHz          | RF 2,4GHz          | RF 2,4GHz          |
| Spotreba               | 100mW                    | 100mW              | 100mW              | 1mW                |
| Dátová priepustnosť    | 16Mbps                   | 1Mbps, 2Mbps       | 1Mbps, 2Mbps       | 1Mbps              |
| Dosah                  | 1 meter                  | 1 meter            | Typická domácnosť  | 3 metre            |
| Podporované zariadenia | 2                        |                    | 127                | 8                  |
| Hlasové kanály         | 1                        | VoIP               | 6                  | 3                  |
| Adresovania            | 32 bit physical ID       | 48 bit MAC         | 48 bit MAC         | 48 bit MAC         |

Tabuľka 4: Porovnanie bezdrôtových technológií prevzaté z [13]

Napriek tomu že bol Bluetooth pôvodne navrhnutý kvôli odstráneniu káblov RS-232 spájajúcich rôzne zariadenia, došlo k jeho veľkému rozvoji na poli mobilných zariadení a počítačov. Dnes už nie je nič neobvyklého na tom, keď si dvaja ľudia vymieňajú aplikácie, obrázky alebo mediálny obsah pomocou Bluetooth, alebo bezdrôtové tlačenie dokumentov na tlačiarni podporujúcej túto technológiu. Bluetooth je veľmi zaujímavá technológia, ktorá má určite budúcnosť.

Dôležitou otázkou pri vytváraní sieťových hier je aj cenová relácia pripojenia. Nie každý užívateľ má dosť peňazí na to aby si mohol zaobstarať prístupový bod (access point, na niektorých zariadeniach len AP) na WiFi prepojenia. A aj použitie technológií v samotných telefónoch predražuje tieto zariadenia.

## 2.5 J2ME a obmedzenia mobilných telefónov

Platforma Java Micro Edition (ďalej len JavaME) je najrozšírenejšou platformou medzi mobilnými telefónmi. Preto bola ako programovací jazyk práce zvolená Java 2 Micro Edition (J2ME). Taktiež dostupnosť vývojových prostredí a aplikácií hovoria v prospech tohto jazyka.

### 2.5.1 JavaME a hardvérové obmedzenia s tým spojené

J2ME je určená pre mobilné zariadenia, ako PDA alebo mobilné telefóny. S tým sú spojené určité hardvérové obmedzenia. Na rozdiel od normálnych Java aplikácií, ktorých behovým prostredím je Java Virtual Maschine (ďalej len JVM), aplikácie pre mobilné zariadenia napísané v J2ME bežia na Kilobyte Virtual Maschine (ďalej len KVM).

Java 2 Micro Edition sa delí na konfigurácie a profily. Pre mobilné telefóny je určená Connected Limited Device Configuration (ďalej len CLDC). Táto konfigurácia špecifikuje základné triedy a rozhrania, ktoré môže programátor aplikácií pre mobilné telefóny použiť a zariadenie musí pomocou KVM vedieť vykonať. Špecifikácia CLDC definuje mobilné zariadenia, ako zariadenie s:

- celkovou pamäťou 160 – 512 KB vyhradenou pre J2ME aplikácie.
- 16 – 32 bitovým procesorom
- nepretržitým sieťovým spojením
- nízkou spotrebou

Tieto nízke hardvérové nároky obmedzujú možnosti mobilných aplikácií a o to viac video hry bežiacie v reálnom čase (ďalej len real-time hry).

Profily v J2ME upresňujú softvérové a hardvérové nároky, ktoré musia mobilné aplikácie zohľadňovať. Tieto profily sa súhrnne nazývajú Mobile Information Device Profile (ďalej len MIDP). Momentálne existujú tri verzie 1.0, 1.1, 2.0.

Hardvérové nároky definované MIDP sú:

- display s 1-bitovou farebnou hĺbkou (čierno-biela) a rozlíšením 96 na 54 pixelov
- jednoručná klávesnica alebo obojručná QWERTY klávesnica
- 128 kB pamäte pre MIDP, 8 kB pre trvalé dáta aplikácie a 32 kB pre KVM
- Obojsmerné bezdrôtové sieťové spojenie

Medzi softvérové nároky patria VM s jadrom pre správu hardvéru a vlákien, musí byť dostupný mechanizmus pre čítanie a zápis do pamäte a do sieťového pripojenia. Je tiež vyžadovaný softvér pre tzv. life-cycle aplikácie, teda spravujúci beh aplikácie od spustenia, cez beh a prerušenia až po koniec aplikácie.

KVM je odľahčená verzia JVM s obmedzenou funkcionalitou a programátorskými možnosťami. Nemá ClassLoader a balíčky prevzané z JavaSE neobsahujú všetky triedy. Navyše ani triedy ňou podporované, majú niektoré metódy implementované len čiastočne. MIDP a CLDC síce špecifikujú, aké triedy má KVM poznať, ale už nie ako majú byť implementované. Implementácie KVM sa preto na rozličných zariadeniach líšia. Napríklad, implementácia metódy

setFullScreenMode() triedy Canvas u zariadení firmy Motorola umožňuje využiť celú obrazovku zariadenia, kdežto u zariadení firmy Nokia ostáva horný pás displaya, obsahujúci indikátor signálu a stavu batérie pre programátora nedostupný.

### 2.5.2 Java a krátko dosahové bezdrôtové spojenia.

J2ME poskytuje krátkodosahové spojenie len jedno API. Je ním JABWT popísané v JSR-82. Jedná sa o API pomocou ktorého je možné pre klientské zariadenie vytvoriť službu (service) a pre zariadenie s funkciou master vyhľadávať dostupné klientské služby, filtrovať ich a pripojiť sa na vybrané z nich.

Pre IrDA ani WiFi v J2ME neexistuje žiadne API. To ale ešte neznamená, že týmito technológiami neje možné pomocou Javy spojenie nadviazať. Čo ale J2ME umožňuje je socketové spojenie. Pomocou toho ide vytvoriť klient-server spojenie využívajúce sieťový protokol TCP/IP, je už potom na zariadení, akú technológiu pre toto spojenie ponúkne (teda napríklad u telefónov Nokia vyšších rád je to aj WiFi).

Ďalej J2ME podporuje pripojenie simulujúce sériové spojenie. To je u niektorých telefónov možné nastaviť aby simulovalo IrDA spojenie.

# Kapitola 3

## Analýza problému

### 3.1 Popis problému

Hlavným cieľom tejto práce je naprogramovať real-time video hru MobilBomber s podporou multiplayer. Keďže ešte v súčasnosti je spojenie cez mobilný internet drahé, práca bude zameraná na bezplatné technológie. Takýmito technológiami sú infračervené spojenie IrDA, Bluetooth a bezdrôtový ethernet WiFi. Hra samotná a realizácia sieťového prepojenia by mali byť navzájom nezávislé. Jadro sieťového spojenia by teda malo byť použiteľné aj pre iné hry a programy. Je treba vymyslieť jednotné rozhranie pre komunikáciu programu (v našom prípade videohra MobilBomber) so sieťovým modulom. Tým by mohlo byť neskôr, v prípade príchodu novej sieťovej technológie, možné rozšíriť sieťové možnosti programu (hry) s minimálnymi zásahmi do jej zdrojového kódu. V rámci sieťovej komunikácie je treba vyriešiť vzájomné vyhľadanie zariadení pri použití danej technológie. Preskúmať hrateľnosť a v čom je nevýhodnejšia oproti iným technológiám. Z programátorského hľadiska je potrebné vyriešiť aj hru viac ako dvoch hráčov.

Pôvodná myšlienka spoločnosti Sun Microsystems bola maximálna prenositeľnosť programov napísaných v jazyku Java. Preto bude hlavným cieľom naprogramovať hru a sieťové pripojenie tak, aby bola zaručená čo najvyššia portabilita. Teda aby bol program (hra) spustiteľný a použiteľný na čo najviac mobilných zariadeniach.

### 3.1.1 Komunikačná knižnica

Komunikačná knižnica by mal byť samostatne stojaci balíček nezávislý na hre samotnej. Mal by obsahovať jednotné rozhranie pomocou ktorého by jadro hry mohlo pristupovať k sieťovému spojeniu, bez toho aby vedelo aká technológia bola použitá. Jediný kontakt hernej časti s konkrétnymi bezdrôtovými technológiami by mohol byť v menu pri výbere spojenia. Pridanie novej komunikačnej technológie, by teda znamenalo pridanie a nastavenie novej položky v menu a vytvorenie objektu novej technológie.

Na rozdiel od technológie WiFi (kde pomocou protokollu TCP/IP sa klienti pripojujú na službu poskytovanú serverom), u technológie Bluetooth pri tvorbe siete pozostávajúcej z viac, ako dvoch zariadení sa zariadenie, nazývané v anglickej literatúre master, pripája k jednému až siedmim Bluetooth zariadeniam. Tieto zariadenia, ktoré sa nazývajú slave, poskytujú svoju službu masterovi. Toto všetko sa deje na sieťovej úrovni. Je preto nutné, pre univerzálnosť komunikačnej knižnice a nezávislosť hry na dané spojenie, vytvoriť na aplikačnej úrovni jednotnú logiku server - klient pre všetky použité (a prípadne aj budúce) bezdrôtové spojenia.

Aplikácia využívajúca túto knižnicu teda musí mať dva módy. Jeden mód bude vyžadovať, aby boli spustení najprv klienti a potom server. Druhý mód bude vyžadovať najprv spustenie serveru a až potom klientov.

Pretože má (u technológie Bluetooth) master na rozdiel od zariadenia slave schopnosť pripojiť viac zariadení a zariadenie slave neumožňuje pripojiť viac zariadení master [14], mal by na aplikačnej úrovni plniť funkciu serveru.

### 3.1.2 Hra MobileBomber

MobilBomber je akčná hra v reálnom čase určená pre viacerých hráčov. Každý hráč ovláda jednu postavičku, bombera. Jeho cieľom je zničiť postavičky súperov a pritom ostať nažive. K ničeniu súperov používa jednu alebo viacero bômb, ktoré ukladá na zem. Tie vybuchujú automaticky po určitej časovej dobe. Postavička hráča je zničená, ak ju zasiahne plameň bomby a to ako súperovej, tak aj vlastnej.



#### Začiatok hry:

Na začiatku hry sa zobrazí hracia plocha. Na ploche sa objavia postavičky všetkých hráčov, ktorý sa prihlásili do hry. Každá postavička má len jednu bombu, ktorej dosah (dĺžka plameňa, ktorý z nej pri výbuchu vyšľahuje) je jeden dielik v horizontálnom aj vertikálnom smere. Okrem nezničiteľných stien sa na ploche nachádzajú aj steny zničiteľné, v ktorých môžu byť ukryté ďalšie bomby a ich vylepšenia.

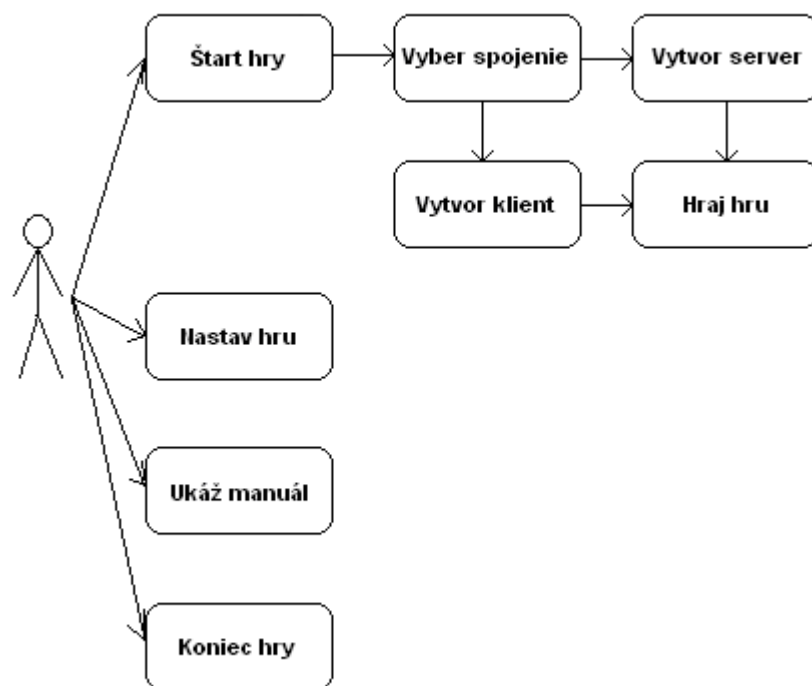
#### Priebeh hry:

Počas priebehu hry postavičky odpaľujú zničiteľné steny, zbierajú nájdené bomby a vylepšenia. Takto si razia cestu k svojim súperom. V prípade stretu so súperovou postavičkou snažia sa ju obkľúčiť svojimi bombami. Čím viac bômb postavička nesie, tým je väčšia pravdepodobnosť, že zo vzájomného stretu so súperom vyjde víťazne. Na druhej strane je ale náročnejšie rozmiestniť tieto bomby tak, aby nebola ohrozená aj postavička samotného hráča.

#### Koniec hry:

Hra končí, ak na hracej ploche zostane už len jedna alebo žiadna postavička. Hráč vyhráva, ak je jeho postavička posledná živá na hracej ploche. Remíza nastáva, ak po výbuchu bomby už nezostane nažive žiadna postavička. Hráč prehráva, ak vyhral nejaký jeho protihráč.

Nasledujúci Use case diagram zobrazuje cielené použitie programu:



**Obrázok 4: Diagram prípadov použitia**

*Štart hry* – umožní užívateľovi nastaviť typ bezdrôtového spojenia

*Nastav hru* – umožní užívateľovi nastaviť základné herné vlastnosti, ako napríklad počet hráčov

*Ukáž manuál* – ukáže užívateľovi základný návod na ovládanie hry

*Ukonči hru* – ukončí aplikáciu a vráti hráča naspäť do menu

*Vyber spojenie* – dá možnosť užívateľovi vybrať z dostupných možností bezdrôtového spojenia (v práci budú implementované WiFi a Bluetooth)

*Vytvor klient* – vytvorí pasívneho, alebo aktívneho klienta

*Vytvor server* – umožní vytvoriť hru na ktorú sa klient pripojí

*Hraj hru* – spustí hru

### 3.1.3 Požiadavky na hru a komunikačný protokol

Hra je typu real-time (teda beží v reálnom čase) a vyžaduje neustálu sieťovú komunikáciu všetkých zúčastnených strán. Ku všetkým hráčom sa musí neustále dostávať informácia o polohe živých postavičiek a informácia o polohe a čase polozenia bomby, aby bolo možné vypočítať prípadný zásah postavičky. Informácia o zničených stenách a hlavne ktoré vylepšenia už boli súperom nájdené a ktoré nie, aby nedochádzalo k situácii, že pre zlú synchronizáciu zoberú dvaja hráči to isté vylepšenie. Ďalej, aby sa nestalo, že na zariadení u jedného hráča je postavička zasiahnutá plameňom a mala by zomrieť a u druhého hráča stihla postavička plameňom uniknúť.

Potrebné je vyriešiť výbuchy bômb. Bomba by mala byť samostatná jednotka (napríklad objekt), ktorá sama po istom čase vybuchne a vypočíta, ktoré steny a ktorí hráči boli zasiahnutí. Nemalo by to byť teda v réžii hráča, keďže aj on samotný môže byť zasiahnutý svojou vlastnou bombou. Na druhej strane nemusí byť ani väzba medzi postavičkou a bombou, keďže víťaz sa určuje na základe toho, kto prežil.

Komunikačný protokol musí byť vymyslený tak, aby minimalizoval objem prenesených dát a tým znižoval efekt latencie sieťového pripojenia. Dôležitá je jeho jednoduchosť a prehľadnosť, aby bol čo najviac univerzálny a mohol byť použitý pri podobných hrách, prípadne, aby bolo možné naprogramovať nový sieťový modul pri objavení novej sieťovej technológie.

## 3.2 Komunikačné technológie podporované hrou MobilBomber

Technológií bezdátového spojenia mobilných telefónov je v súčasnosti niekoľko. Každé má svoje výhody a nevýhody. Ide ich ale rozdeliť do niekoľko

kategórií. Základným kritériom je pritom maximálna vzdialenosť spojených zariadení, cena prenosu a latencia, teda rýchlosť odozvy.

Podľa toho je možné ich rozdeliť do dvoch základných skupín.

Ďalekodohové bezdrôtové technológie sú väčšou kategóriou. Vznikli skôr a boli vyvinuté pre potreby telefónov. Pre komunikáciu využívajú mobilnú telefónnu sieť. Ich výhoda je dosah. Spojenie je totiž možné všade tam, kde je pokrytie telefónnej siete. Na druhej strane, pri výpadku tejto siete nie je možné tieto technológie využívať. Výraznou nevýhodou je ich vyššia latencia a cena. Ďalšou nevýhodou bola aj prenosová rýchlosť. V súčasnosti sú už ale vo vývoji technológie, ktoré budú konkurovať drôtovému širokopásmovému spojeniu [18]. Patria medzi ne GPRS, EDGE, atd...

Krátkodosahové bezdrôtové technológie neboli pôvodne určené na komunikáciu medzi mobilnými telefónmi. Ich dosah sa pohybuje v niekoľkých metroch. Rýchlosti prenosu sú pre svet mobilných telefónov dostačujúce. Odozva je lepšia ako u ďalekodohových technológií. Sú nezávislé na mobilnej sieti, teda v prípade jej výpadku sú stále funkčné. Ďalším veľkým pozitívom je, že dátový prenos je bezplatný. Implementácia prijímacieho a vysielačieho zariadenia siete pridáva a cene mobilného telefónu, ale nijako moc ho nepredražuje [18].

Medzi tieto technológie patria napríklad Bluetooth, WiFi a IrDA.

Multyplayer hry by sa podľa Nokie [14] dali rozdeliť do hlavných štyroch kategórií.

- Snímkovo orientované hry. Sú to hry, pri ktorých dochádza ku výmene dát vždy pri vykreslení jedného snímku. Tento spôsob je síce najbezpečnejší. Nepochádza pri ňom k nepredvídanému pohybu postavičiek. Na druhej strane, ale požaduje čo najnižšiu latenciu. Tá by mala byť maximálne 40 milisekúnd, keďže na simuláciu plynulého pohybu potrebujeme dosiahnuť aspoň 25 snímok za sekundu.
- Hra založená na rozoznávaní smrti. Vykresľovanie snímok a sieťový prenos sú vykonávané nezávisle na sebe. Pri vypadnutí spojenia s jedným klientom sa vypočíta jeho predpokladaná poloha. Tá sa pri obnovenom spojení opraví. Výpadok spojenia nastáva na zlomok sekundy, takže odchýlka od reálnej pozície protihráča nie je veľká a je okamžite opravená.

- Synchronná simulácia. Vykresľovanie snímok a sieťový prenos sú taktiež vykonávané nezávisle na sebe. Pri výpadku alebo omeškaní spojenia jedného klienta master (server) obnoví ostatných klientov až keď mu príde informácia od tohto klienta. Latencia sa pohybuje v stovkách sekúnd.
- Ťahová hra. Master (server) čaká, až kým klient, ktorý je na ťahu, nepošle svoje informácie. Potom obnoví obrazovku a predá ťah druhému hráčovi. Tento typ sieťovej hry dovoľuje latenciu aj pár sekúnd, kdeže hráči zvyknú dlhšie premýšľať. Medzi tieto hry patrí napríklad šach alebo piškvorky.

Z charakteru hry MobilBomber vyplýva, že do úvahy pripadajú prvé dva typy. Tie vyžadujú pomerne nízku latenciu. Preto musí byť vybraná sieťová technológia rýchla. Túto požiadavku splňujú práve krátkodosahové bezdrôtové komunikačné technológie. Jedine spojenie cez IrDA má časté výpadky a dovoľuje prepojiť len dve zariadenia. Jeho dosah je len 1-2 metre a nízka prenosová rýchlosť obmedzuje možnosti prenosu väčšieho množstva informácií [19]. Preto nie je IrDA vhodné na tvorbu multiplayer real-time hier pre mobilné telefóny. Bez problémov ale môže byť použité na tvorbu ťahových hier.

|                      | WiFi       | Bluetooth  | IrDA                          |
|----------------------|------------|------------|-------------------------------|
| Odozva pri pripojení | 2 sec      | 10 sec     | 250 – 650 ms                  |
| Komunikačná odozva   | < 10 ms    | 1.2 ms     | < 20 ms, môže padnúť na 500ms |
| Dátová priepustnosť  | 4,5 Mbit/s | 400 Kbit/s | 80 Kbit/s                     |

**Tabuľka 1** prevzatá z [19]

### 3.2.1 Bluetooth

Technológia Bluetooth [18][19] bola povodne určená na prepojenie mobilného telefónu s periférnymi zariadeniami(napr. HandsFree). Neskôr bola pridaná aj možnosť komunikácie dvoch mobilných zariadení a umožňuje dva typy spojenia. Synchronous connection oriented (SCO) a Asynchronous connection-less (ACL). SCO je určená na prenos zvuku. Nás bude zaujímať ACL, ktorá je určená na prenos dát. ACL využíva dva druhy paketov. DM a DH. DM využíva technológiu forward error correction (FEC) a opravu chýb vzniknutých pri prenose. DH pakety FEC nevyužívajú a preto môžu prenášať väčší objem dát. Číslo za názvom paketu označuje časový slot, ktorý je u Bluetooth 625 mikro sekundy.

| Typ | Sloty | hlavička[b] | dáta[b] | FEC  | sym   | asym↑ | asym↓ |
|-----|-------|-------------|---------|------|-------|-------|-------|
| DM1 | 1     | 1           | 0-17    | 2/3  | 108.8 | -     | -     |
| DH1 | 1     | 1           | 0-27    | none | 172.8 | -     | -     |
| DM3 | 3     | 2           | 0-121   | 2/3  | 258.1 | 387.2 | 54.4  |
| DH3 | 3     | 2           | 0-183   | none | 390.4 | 585.6 | 86.4  |
| DM5 | 5     | 2           | 0-224   | 2/3  | 286.7 | 477.8 | 36.3  |
| DH5 | 5     | 2           | 0-339   | none | 433.9 | 723.2 | 57.6  |

Tabuľka 2 prevzatá z [14]

### 3.2.2 WiFi

WiFi je technológia na prepájanie počítačov a iných zariadení a vzdialenosť 100m. Niekedy sa jej preto hovorí Wireless LAN. Ako bolo vidieť na tabuľke 1, relatívne dobrá latencia a vysoká prenosová rýchlosť jej dávajú dobré predpoklady na rýchlu komunikáciu a teda aj programovanie real-time multiplayer hier. Od

svojho vzniku prešla WiFi dlhým vývojom a dnes existuje viac štandardov. Štandardy IEEE 802.11b a 802.11g majú maximálnu prenosovú rýchlosť 54mb/s. Ich cena je nižšia ako u štandardu 802.11a preto sú viac používané v domácich počítačových sieťach a v zariadeniach určených pre každodenné použitie (ako napríklad PDA, mobil). Čo je dôležité, sú navzájom kompatibilné. Je teda možné pripojiť na acces point štandardu IEEE 802.11g zariadenie podporujúce 802.11b.

Preto sú tieto dva štandardy pre programovanie mobilných hier zaujímavé a budeme sa nimi ďalej zaoberať.

### 3.2.3 Zhodnotenie

Z hore uvedenej analýzy vyplýva, že Bluetooth a WiFi sú vhodné na prenos, kdežto IrDA nevyhovuje. IrDA bolo počas písania práce testované aj na hre Snake, ktorá je popísaná v kapitole 7. Tam sa ukázalo aj v praxi, že pre real-time multiplayer hry je táto bezdrôtová technológia nepoužiteľná. Infračervené spojenie nebude teda implementované v hre MobilBomber. Tento typ komunikácie ale bude zahrnutý do komunikačnej knižnice pre možnosť použiť IrDA v ťahových hrách.

# Kapitola 4

## Návrh riešenia

Výsledné riešenie bude pozostávať z dvoch samostatných častí. Jednou bude komunikačný modul, ktorý bude zabezpečovať komunikáciu medzi zariadeniami cez Bluetooth a WiFi. Druhou samostatnou časťou bude multiplayer hra MobilBomber, určená pre divoch až štyroch hráčov. Tieto časti musia byť na sebe nezávislé, aby pre hru bolo možné naprogramovať aj iný sieťový modul bez potreby zasahovať do jej zdrojového kódu. Taktiež by existujúce moduly mali byť použiteľné na programovanie nových multiplayer sieťových hier. Preto je nutné navrhnuť kvalitné, čo najuniverzálnejšie programátorské rozhranie.

### 4.1 Rozhranie

Rozhranie by malo definovať interný komunikačný protokol, akou „rečou“ budú medzi sebou zariadenia komunikovať. Pre odľahčenie dátového spojenia a na zmiernenie prejavov jeho latencie nebudú medzi zariadeniami posielané súradnice a zvlášť bomby, ale len jednobajtové správy o zmene stavu postavy. Zmena stavu bude obsahovať zmenu smeru a jeden bit správy bude rezervovaný pre informáciu o tom, či počas pohybu (zmeny smeru) došlo k položeniu bomby. Oproti súradnicovom prístupe je síce tento prístup šetrnejší na prenosovú rýchlosť, ale vyžaduje navyše prepočítavanie nového stavu.



- Na začiatku hry sa vymenia tzv. HandShake správy.
- Potom sa medzi zariadeniami vymenia správy o štartovných pozíciách hráčov na mape.
- Správy preposielané počas hry budú obsahovať aktuálnu zmenu stavu (smer/bomba) objektov.

## 4.2 Komunikačný modul

Bude pozostávať z jednej alebo viacerých tried. Jeden alebo viac druhov spojenia budú zjednotené pomocou univerzálneho rozhrania, ktoré budú implementovať všetky triedy daných troch spojení.

## 4.3 Hra MobilBomber

Hracia plocha bude mať charakter šachovnice. Na obrazovke bude vykreslená celá časť hracej plochy. Pri pohybe hráča sa bude prekresľovať aj, pre hráča viditeľná, časť hracej plochy.

Hra bude rozdelená podľa Model-View-Controller architektúry. Vrstva Model bude mať za úlohou správu a zmenu stavu herných objektov, ako je bomba, postava hráča alebo herný plán. Ďalej je potrebná vrstva Controller zaručujúca spracovanie príkazov zo vstupu. Vstupom bude klávesnica mobilného zariadenia, sieťový vstup pre bezdrôtové technológie, alebo generátor náhodného pohybu, prípadne sofistikovanejšieho algoritmu UI. Vykresľovanie herného plánu a ostatných objektov bude mať na starosť View vrstva. Tá bude spolupracovať s displayom zariadenia a vykresľovať zmeny stavu Model vrstvy.

Každej z týchto vrstiev (častí) bude odpovedať jeden Java balíček (package), ktorý bude implementovať dané funkcionality.

Pohyb hráča bude riadený pomocou klávesnice mobilného telefónu. Postavička ovládaná hráčom sa pritom bude môcť pohybovať len vo vodorovnom a zvislom smere, nie po diagonálach, a to len po poličkách šachovnice, ktoré sú „voľné priechodné“. Voľne priechodné políčko je pritom prázdne políčko, alebo políčko na ktorom sa nachádza nejaký predmet. Nepriechodné políčko zahŕňa steny rozbitné aj nerozbitné a aktivované bomby. Presun z jedného políčka na druhé je pritom spojený a nie po skokoch.

# Kapitola 5

## Užívateľská dokumentácia

### 5.1 Inštalácia programu

Inštalácia MIDletu, tak sa volá mobilná aplikácia - resp. program - vôbec nie je náročná. Človek nemusí byť programátorom, aby spojzndnil mobilnú aplikáciu na svojom telefóne. Existuje viac spôsobov ako dostať program do telefónu:

1. Stiahnuť z počítača pomocou kábla alebo Bluetooth.
2. Stiahnuť z internetu (ak to váš telefón podporuje a máte zaplatenú príslušnú službu).
3. Stiahnuť z iného telefónu pomocou Bluetooth.

To ako stiahnutie vykonať sa dozviete z návodu, ktorý ste dostali spolu s vašim telefónom pri kúpe.

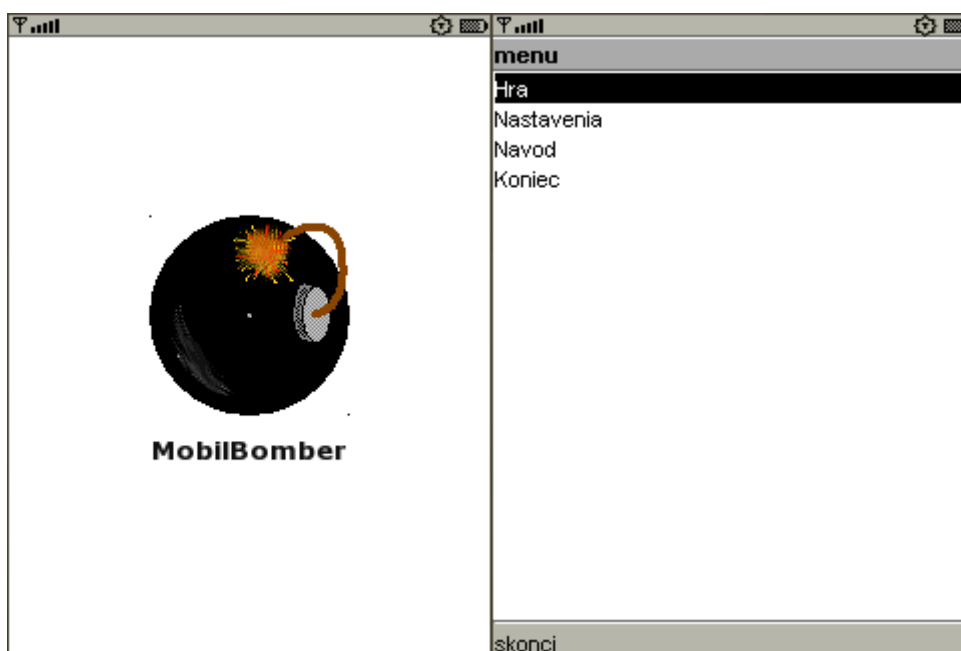
Dôležité ale je, čo sťahovať. Mobilná aplikácia (MIDlet) je zabalená spolu s inými pomocnými súbormi v balíčku Java archive (ďalej JAR). Súbor teda bude mať názov *nazov\_saboru.jar*. Spolu s týmto súborom býva distribuovaný aj súbor *nazov\_saboru.jad*, ktorý má ale len informatívny charakter a pre inštaláciu aplikácie a jej používanie nie je potrebný.

Po stiahnutí JAR súboru do pamäti mobilného telefónu je potrebné ho nainštalovať. Z pravidla býva JAR súbor zaradený mobilným telefónom medzi nenainštalované súbory. Tento súbor nájdeme a voľbou v menu nainštalujeme do telefónu. To sa ale môže líšiť u rôznych modeloch a značkách telefónov. Preto pre

podrobnejší postup inštalácie už stiahnutého programu, ktorý je u všetkých aplikácií rovnaký, použite návod dodaný k telefónu.

## 5.2 Ovládanie a pravidlá hry

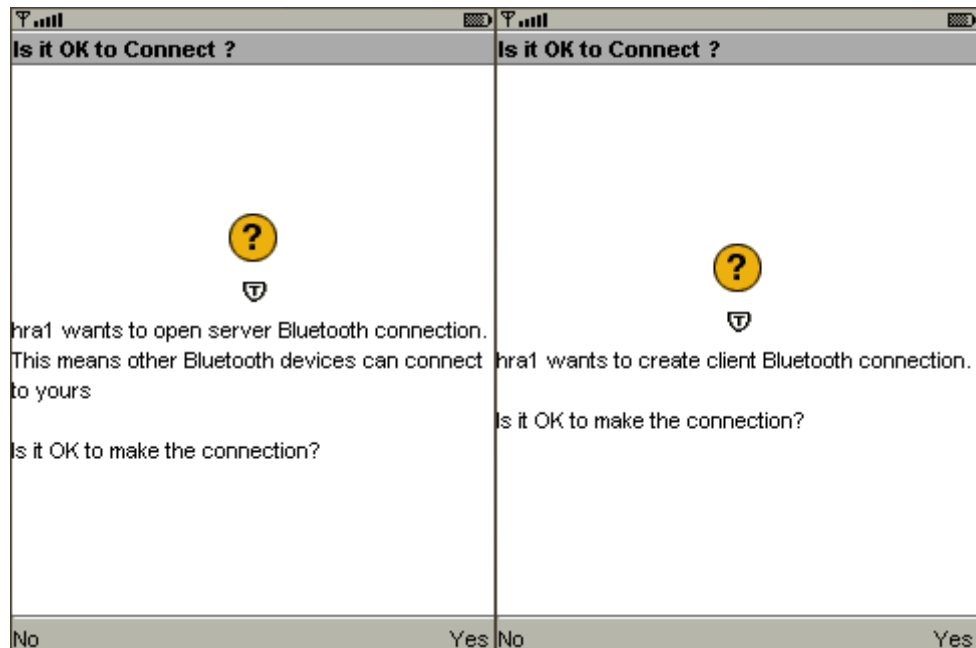
Po inštalácii Java aplikácie do mobilu je štandardne aplikácia zaradená do zoznamu alebo zložky inštalovaných aplikácií. Pre spustenie aplikácie je nutné ho nájsť. Väčšinou je dostupný priamo v hlavnom menu telefónu. Spustenie aplikácie prevedieme kliknutím hlavného tlačidla joysticku. V prípade, že telefón nereaguje, nájdete bližšie informácie v návode, dodanom k mobilnému telefónu. Ak sa aplikáciu podarilo spustiť, mala by sa objaviť úvodná obrazovka a vzápätí po nej aj hlavné menu hry.



Obrázok 5: Úvodná obrazovka a hlavné menu

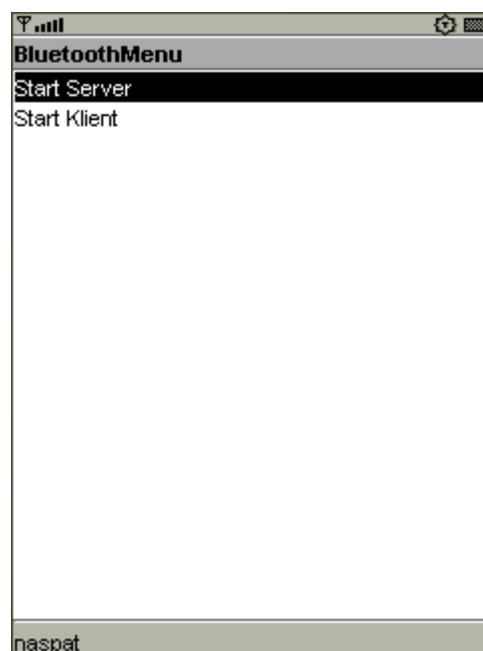
Po spustení aplikácie zvolíme typ sieťového pripojenia. Teda, či zariadenie bude plniť funkciu serveru alebo klienta (na aplikačnej úrovni, čo na fyzickej úrovni odpovedá úrovniam master a slave). V prípade pripojenia cez WiFi, ak mobilné

zariadenie nie je práve pripojené k acces point-u, telefón nás upozorní, že aplikácia chce naviazať bezdrôtové spojenie cez rozhranie WiFi.

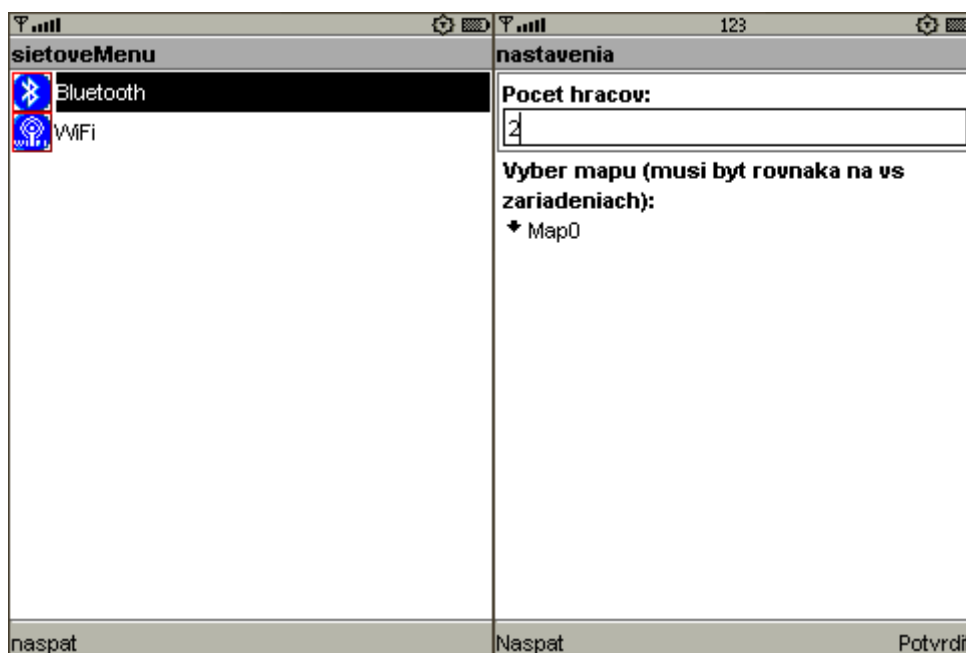


Obrázok 6: Oznámenie o pokuse pripojiť sa cez Bluetooth

Pri spojení Bluetooth je nutné, aby boli spustené najprv všetky klientské aplikácie a až potom server. Je to teda presne naopak, ako u Wifi spojenia.

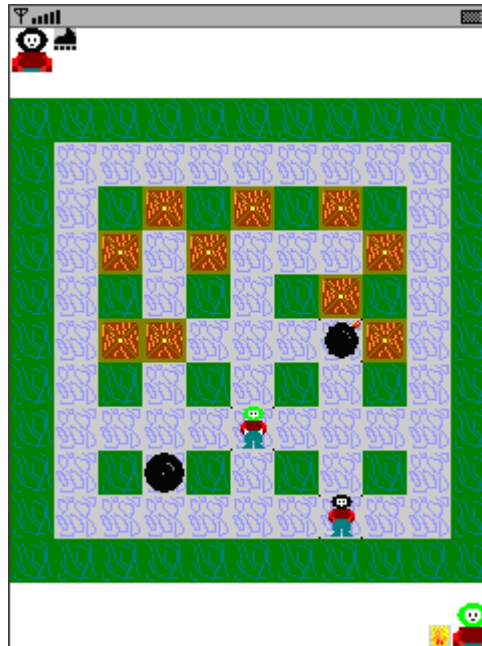


Obrázok 7: Úvodná obrazovka a hlavné menu



Obrázok 8: Siet'ové menu a nastavenia

V prípade, že užívateľ súhlasí, zariadenie nám dá vybrať zo zoznamu dostupných sietí. Po zvolení siete sa aplikácia pokúsi o spojenie s druhým telefónom. Ak sa spojenie podarí, začína hra. V prípade, že z akéhokoľvek dôvodu nebolo možné naviazať spojenie, zobrazí sa úvodná obrazovka a je možné pokúsiť odznova nastaviť pripojenie a spojiť zariadenia.



Obrázok 9: Herná obrazovka

Pre ovládanie postavy hráča sa používajú takzvané herné klávesy. Tie sú na niektorých mobilných telefónoch namapované na joystick alebo smerové tlačidlá. U telefónov, ktoré tieto ovládacie prvky nemajú bývajú herné klávesy namapované na nasledujúce tlačidlá telefónu:

- [2] pohyb hore
- [8] pohyb dole
- [4] pohyb doľava
- [6] pohyb doprava
- [5] polozenie bomby



Bomby:

Postava hráča nesie so sebou jednu až osem bômb. Tieto môže položiť po jednom stláčaním klávesy „položenie bomby“, alebo bezprostredne za sebou stlačením a podržaním tejto klávesy. Vždy ale len toľko, koľko práve nesie.

V prípade, že sú všetky jeho bomby položené, ďalšiu jednu bombu môže hráč položiť, až keď vybuchne nejaká jeho položená bomba.

Pokladanie bômb:

Stlačením príslušného tlačidla dôjde k položeniu bomby. Bomba nevybuchne hneď, ale až po určitom časovom intervale.

Hlavnou úlohou hráča je zabiť protihráčov a zároveň ostať nažive. Tento fakt nie je tak jednoznačný, ako sa na prvý pohľad môže zdať. Často sa totiž stáva, že sa súperia obkľúčia navzájom a zomierajú naraz. Prípadne nestihne útoiaci hráč uniknúť svojej vlastnej bombe. V tomto prípade, keď nezostane nikto živý, nastáva remíza a kolo sa hrá odznova. Ináč sa stáva víťazom posledný hráč, ktorý ostal nažive.

Počas hry môže hráč, prostredníctvom postavy ktorú ovláda, zbierať rôzne predmety ako *bomby*, *plamene*, *korčule*.



*Bomba*: zvýši počet bômb ktoré postava nesie so sebou a môže použiť proti svojim súperom.



*Plameň*: plameň zvyšuje dosah bomby



*Korčule*: zvyšujú rýchlosť postavy



# Kapitola 6

## Programátorská dokumentácia

### 6.1 Štruktúra programu

Program sa skladá z dvoch hlavných častí. Prvá je sieťová knižnica obsahujúca rozhrania pre klientské a serverové spojenia a ich implementácií pre jednotlivé krátkodosahové bezdrôtové komunikačné technológie. Druhá časť je samotná hra, ktorá využíva sieťovú knižnicu pomocou jej univerzálneho komunikačného rozhrania.

### 6.2 WirelessConnectionLibrary

WirelessConnectionLibrary je samostatne kompilovateľný balíček (package) obsahujúci triedy a rozhrania potrebné pre vytvorenie bezdrôtového spojenia. Je možné pomocou neho vytvoriť spojenia cez Bluetooth a WiFi.

#### 6.2.1 Obsah balíčka

Balíček WirelessConnectionLibrary obsahuje nasledujúce triedy implementujúce dané spojenia:

`BluetoothClient` - trieda zodpovedná za inicializáciu BT službu hry na strane klienta

|                                   |  |
|-----------------------------------|--|
| <code>BluetoothMultiServer</code> | - trieda zodpovedná za vyhľadanie BT služby hry inicializovanej klientom a pripojenie k jej odberu |
| <code>WifiKlient</code>           | - trieda zodpovedná za vyhľadanie socket serveru   |
| <code>WifiMultiServer</code>      | - trieda zodpovedná za založenie socket serveru  |

Rozhrania, ktoré definujú, ako má vyzerat' trieda pre server a ako pre klientskú časť spojenia:

|  |                  |
|--|------------------|
| <code>ConnectionInterface</code>       | -definuje server |
| <code>ServerConnectionInterface</code> | -definuje klient |

Obsahuje tiež triedu `IrDaClient` umožňujúcu vytvorit' infračervené spojenie. Tá sa ale v hre nevyužíva, lebo použitie infračerveného spojenia na hry v reálnom čase pre viac hráčov, ako je napríklad aj `MobilBomber`, je veľmi obmedzené a podpora IrDA na nových zariadeniach je malá a ďalej sa znižuje. Toto spojenie je ale použiteľné pre vývoj jednoduchých ťahových hier pre dvoch hráčov a preto bolo v balíku knižnice ponechané.

## 6.2.2 Použitie a inicializácia knižnice

`WirelessConnectionLibrary` je univerzálna knižnica vhodná na všetky typy streamových bezdrôtových spojení. Dá sa teda použiť, ako na prenos dát, tak aj na programovanie ťahových alebo real-time sieťových hier.

Inicializácia spojenia pre Bluetooth nie je zložitá. Pri tvorbe nového projektu je treba najprv nastaviť závislosť projektu na knižnici (konkrétne jej skompilovaného súboru `WirelessConnectionLibrary.jar`) a v kóde importovať balíček

WirelessConnectionLibrary. V prípade tvorby klienta je nutné vytvoriť inštanciu triedy BluetoothKlient. Na to slúži statická singleton trieda ConnectionFactory. Tá pomocou metódy getBTClient() vráti novovytvorenú a inicializovanú triedu Bluetooth spojenia, takže nie je potrebné nastavovať žiadne počiatočné hodnoty. Metódou tejto triedy je openConnection(), ktorá sa postará o otvorenie spojenia a vracia vstavanú triedu jazyka Java pre prúdové spojenie StreamConnection.

Príklad vytvorenia klinetského spojenia:

```
BluetoothClient btClient =  
ConnectionFactory.getInstance().getBTClient();  
StreamConnectoin streamConn = (StreamConnection)  
btClient.openConnection();
```

V prípade tvorby severu je postup podobný, ale pri vytváraní triedy je nutné v konštruktore nastaviť počet spojení, ktoré budeme chcieť vytvoriť.

Príklad vytvorenia multi serverového spojenia:

```
BluetoothServer btServer = (ServerConnectionInterface)  
ConnectionFactory.getInstance().getBTServer();  
StreamConnection[] streamConns = (StreamConnection[])  
serverConn.openConnections();
```

Vytvorenie klientských tried a triedy serveru je postup analogický, ako pri vytváraní Bluetooth spojenia.

Prebieha spojenie cez Bluetooth a WiFi vo vnútri týchto tried popisuje kapitola MobilBomber.

## 6.3 MobilBomber

V tejto kapitole je rozobraná štruktúra tej časti programu, ktorá má na starosť hernú logiku, jej niektoré algoritmy a dátové štruktúry. Nachádza sa tu aj postup zostavenia a prekladu zdrojových kódov.

Hra MobilBomber sa skladá z troch logických modulov. Zobrazovací modul, má na starosť obsluhu a zobrazenie užívateľského rozhrania na obrazovku zariadenia. Ovládací modul má na starosti odchytyvanie kláves a príkazov a ich predávanie hernej logike. Modul entít spravuje postavy hráčov a bomby, ktoré majú tieto postavy pri sebe a môžu kedykoľvek použiť.

Každému z týchto modulov odpovedá jeden Javovský balíček (package).

### 6.3.1 Cyklus hry

Cyklus hry, anglicky tiež game engine, je hlavnou časťou drvivej väčšiny počítačových hier. V hre MobileBomber je implementovaný v metóde `run()` v triedy `Hra`. Nachádza sa v ňom všetko čo je nevyhnutné pre beh hry a jej vykreslenie na obrazovku. Nachádza sa v časti `hra` a je implementovaný v metóde `run()` triedy `GameEngine.java`. Ide vlastne `while` cyklus bežiaci v samostatnom vlákne. Úlohou `while` cyklu je nepretržite zabezpečovať nasledujúce úkony:

1. obsluha kláves
2. aktualizácia stavu hry cez sieť
3. vykreslenie
4. nastavenie doby spánku

Obsluha kláves je zabezpečená metódou `getKeyState()`. Na základe jej návratovej hodnoty sa rozhodne, ktorým smerom sa vydá postava hráča, prípadne či sa má zároveň položiť aj bomba. Táto udalosť vyústi k zmene stavu hry.

Aktualizácia stavu hry má na starosť synchronizáciu mobilných telefónov. V každej iterácii herného cyklu posiela informácie o zmene stavu ostatným zariadeniam.

Vykreslenie hry nastáva po synchronizácii všetkých zariadení zúčastnených v hre. Herná plocha ako aj jednotlivé postavy a bomby sú na obrazovku mobilného telefónu vykreslené prostredníctvom triedy `GameCanvas` profilu MIDP, ktorá sa objavila až vo verzii 2.0.

Nastavenie doby spánku je dôležitou časťou herného cyklu. Zabezpečuje totiž plynulosť hry a rovnomerné vykresľovanie hry na obrazovku.

### 6.3.2 Aktualizácia stavu hry

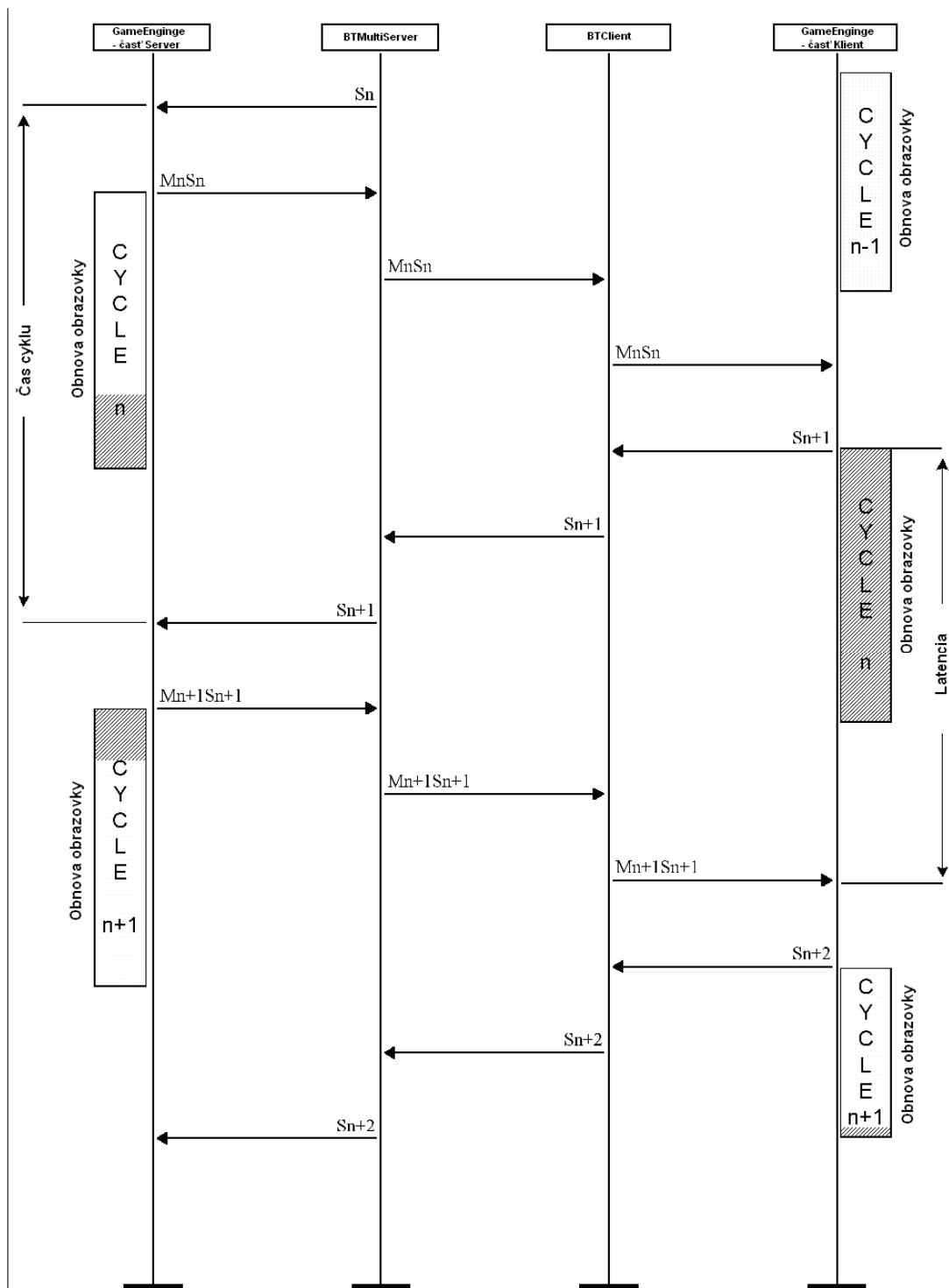
Táto časť je nosnou časťou sieťovej hry. Obrázok 5 ukazuje spojenie serveru (hostu) a klienta resp. v sieťovej terminológii spojenie master a slave. Obe aplikácie majú okrem vlákna aplikácie aj vlákno sieťového spojenia (v tomto prípade ide o Bluetooth spojenie).

Herný cyklus serveru začína prijatím informácií od klienta (Sn). Následne server vyhodnotí svoj herný stav a spolu s informáciami o stave klienta, ktoré od neho obdržal, ich posiela späť klientovi (MnSn). Na základe prijatých informácií vykreslí hernú obrazovku. Následne opakuje cyklus.

Vo vlákne Bluetooth server dochádza k prenášaniam paketov, odoslaných z aplikačného vlákna serveru, do vlákna Bluetooth spojenia konkrétneho klienta. Tieto pakety obsahujú informácie o servery aj klientovi. Naopak pri zasielaní informácií od klienta, vlákno Bluetooth spojenia posiela len informácie o svojom stave.

V cykle klientskej aplikácie prijme klient informácie o stave servera, ako aj svoje vlastné informácie. Následne vyšle nové informácie o zmene stavu. Potom vykreslí hernú plochu spolu s postavami hráčov a bomby na obrazovku mobilného zariadenia. Tým je cyklus dokončený a môže nasledovať jeho nová iterácia.

Tento koncept je inšpirovaný konceptom Nokie [14] a je prispôsobený pre potreby hry `MobileBomber`.



Obrázok 5: Pribeh aktualizácie hry podľa diagramu spoločnosti Nokia prezvané z [14]

### 6.3.3 Vykreslenie na obrazovku

Vykreslenie na obrazovku prebieha v samostatnom vlákne, aby neboli spomaľované iné časti programu. Pomocou triedy `GameCanvas`, ktorá bola pridaná do profilu MIDP až v druhej verzii, sa urýchlilo a zjednodušilo vykresľovanie na obrazovku. S ňou bola v rámci Game Application Programming Interface pridaná abstraktná trieda `Layer`. Od nej dedia triedy `Sprite` a `TiledLayer`. Trieda `Sprite` bola v programe využitá na vytvorenie pohyblivých grafických objektov reprezentujúcich jednotlivých hráčov. Trieda `TiledLayer` slúži na vytvorenie statického alebo animovaného pozadia. V hre bola použitá na vytvorenie statického pozadia.

Game API obsahuje veľmi užitočnú triedu `LayerManager`, ktorá dokáže efektívne organizovať vrstvy `Layer`. Táto trieda implementuje Z-buffer, ktorý určuje hĺbku jednotlivých vrstiev. Tá presne definuje, ako sa budú vrstvy prekrývať s tým, že prvá vrstva pridaná do `LayerManageru` bude úplne na vrchu a na obrazovke nebude žiadnou vrstvou prekrytá a posledná vrstva bude pomyselné vzadu a teda môže byť úplne zakrytá ostatnými vrstvami. `LayerManager` ďalej implementuje metódu, ktorá vykreslí všetky vrstvy do objektu triedy `Graphics` na základe práve spomínaného Z-buffera. To je veľmi dôležité, lebo prístup k displayu telefónu, pre vykreslenie jednotlivých grafických objektov, zaberá viac času, ako ich vykresľovanie do pamäte telefónu a následné vykreslenie celej hernej scény na jeden raz.

## 6.4 Preklad a zostavenie programu

Program `MobilBomber` bol vyvíjaný v editore `NetBeans`, ktorý je podporovaný samotnou firmou `Sun`, tvorcom jazyka `Java`. Zdrojové kódy pre preklad a zdroje sa nachádzajú v zložke `src`,

Preložiť ich môžeme priamo vo vývojovom prostredí `NetBeans` pomocou voľby `Build Main Projekt`. Konkrétne `Build > Build Main Projekt`. Klávesová skratka pre túto operáciu je `F11`.

Vývojové prostredie NetBeans dovoľuje importovať projekty, vygenerované inými vývojovými prostrediami (ako napríklad Eclipse, WirelessToolkit). Jediné, čo programátor potrebuje sú zdrojové kódy a súbor s príponou .jad. Po zadaných týchto súborov do formulára pre vytváranie projektov z existujúcich zdrojov nám prostredie vytvorí kompletný projekt.

### 6.4.1 Ant a Antenna

Preložiť a zostaviť program je možné aj bez pomoci NetBeans. V zložke projektu sa nachádza súbor buildAntenna.xml. Programom Ant je potom možné na základe tohto súboru preložiť a zostaviť projekt MobilBomber.

Antenna je rozšírením programu Ant o takzvané task-s, ktoré sú zaujímavé pre vývojárov mobilných aplikácií pod platformou JME. Zostavovacie súbory projektov pre mobilné zariadenia, využívajúce toto rozšírenie, sú potom jednoduchšie a prehľadnejšie. Pre jej funkčnosť je nutné uložiť súbor antenna-bin-1.2.1-beta.jar do zložky bin/ nachádzajúcej sa v koreňovom adresári programu Ant.

V súbor buildAntenna.xml je potrebné nastaviť cestu v premennej „wtk.home”. Potom možné súbor spustiť z príkazovej riadky z koreňového adresára

```
ant -lib libs -f buildAntenna.xml
```

projektu pomocou príkazu:

## 6.5 Prepojenie pomocou WiFi

Prepojenie mobilných telefónov cez WiFi pracuje na základe sieťového protokolu TCP/IP. V Jave 2 Micro Edition existuje v profile MIPD 2.0 podpora pre sieťovú komunikáciu cez tento protokol v podobe GCF. V GCF je zahrnutá podpora serverových socketov, bezpečnostného pripojenia, a nízkoúrovňového pripojenia pomocou práve spomínaného protokolu.



V MIDP 1 bola dostupná len podpora pre pripojenie klienta. Klient sa mohol pripojiť k už existujúcemu Socket serveru, ktorý bežal na nejakom výpočtovo silnejšom zariadení. Neskôr sa ale výpočtový výkon mobilných zariadení zvýšil natoľko, že bol do nového štandardu (profilu) MIDP 2.0 zahrnutý aj Socket server. Pomocou neho môže mobilné zariadenie aj otvoriť spojenie. To umožňuje prepojenie dvoch mobilných telefónov medzi sebou. Spojenie medzi mobilnými zariadeniami je ale nutné naviazať cez nejaký acces point, keďže v MIDP 2.0 nie je žiadna podpora pre nastavenie WiFi pripojenia a teda ani nie je možné vytvoriť softvérový acces point alebo naviazať Ad-Hoc spojenie.

Socket server J2ME MIDP 2.0 vytvoríme pomocou metódy *open(...)* triedy *Connector* k tomu určenej. Tá vracia triedu *SocketServerConnection*, ktorá práve reprezentuje socket ponúkajúci pripojenie.

```
//vytvorenie socketu pre server
ServerSocketConnection server =
(ServerSocketConnection) Connector.open("socket://:8888");
```

Po vytvorení socketu chceme naviazať spojenie s potenciálnymi klientami. Metóda *acceptAndOpen()* aktivuje server do vyčkávacieho režimu. To spôsobí zastavenie vlákna Serveru podobné prerušeniu pri čakaní na vstup [Herout]. K znovu oživeniu vlákna dôjde až pri pripojení klienta k nášmu serveru.

Pripojenie klientov k nejakému zariadeniu reprezentuje trieda *SocketConnection*, ktorá bola programátorom dostupná už v profile MIDP 1.

```
SocketConnection client =
(SocketConnection) server.acceptAndOpen();
```

Potom už môžeme prejsť k samotnému prenosu dát pomocou prúdov bajtov (tzv. Stream). Pomocou metódy *openOutputStream* otvoríme výstupný prúd pre zápis do socketu. A pomocou *openInputStream* zase otvoríme vstupný prúd pre čítanie zo socketu.

```
//otvorenie streamov
OutputStream os = client.getOutputStream();
InputStream is = client.getInputStream();
```

Na zápis je najvhodnejšia metóda `write`. V triede `OutputStream` sa nachádzajú tri preťažené metódy tohto typu. Jedna má, ako vstup jeden bajt, druhá umožňuje posielanie celého poľa bajtov a posledná umožňuje poslať vybranú časť tohto pola, definovanú na základe druhého a tretieho parametru. Trieda `InputStream` obsahuje tri preťažené metódy `read()`, ktoré umožňujú načítať zo vstupu hodnoty typu `integer`, `byte` a `byte[]`.

```
//samotne citanie a zápis pomocou prúdov
int vstup;
vstup = is.read ();
byte vystup;
os.write(vystup);
```

Po ukončení práce s prúdmi je nutné ich uzavrieť. Taktiež je nutné uzavrieť otvorené sockety, aby boli opäť voľné pre ďalších záujemcov.

```
//uzatvorenie streamov a socketov
is.close();
os.close();
client.close();
server.close();
```

## 6.6 Prepojenie pomocou Bluetooth

Obrázky č. 10 a 11 graficky zobrazujú jednotlivé kroky pri pripájaní telefónu k inému zariadeniu pomocou technológie Bluetooth. Po spustení programu dochádza k vyhľadávaniu zariadení s bežiacim BT Messenger serverom. Pokiaľ je vyhľadávanie

úspešné, môžeme sa ihneď k serveru pripojiť, pokiaľ nie, musíme server na našom zariadení spustiť a čakať, až sa k nám pripojí nejaký účastník. Ako náhle je vyriešené naviazanie spojenia medzi zariadeniami, môže začať ich vzájomná komunikácia. Pokiaľ sa nejaký z účastníkov rozhodne hru ukončiť, je o tom jeho kolega informovaný. Potom môže byť aplikácia riadne ukončená.

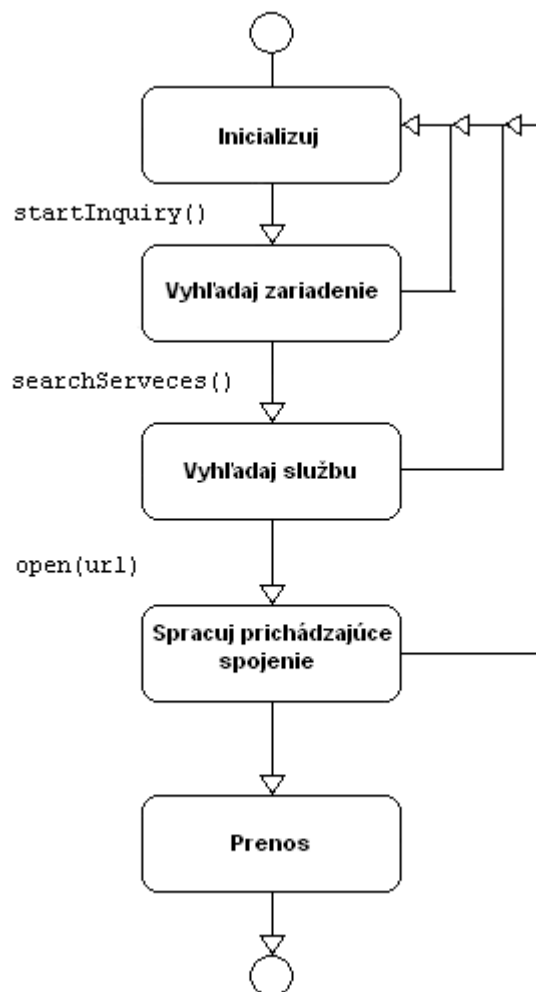
### 6.6.1 Beh serveru

Obrázok č. 10 popisuje beh programu serveru. Podmienka pre popis tohto behu

je, že aplikácia serveru má prístup k zariadeniu Bluetooth a má overené povolenie so zariadením pracovať.

Pri štarte programu zariadenie inicializuje Bluetooth stack, následne server hľadá všetky dostupné zariadenia. Po výbere niektorého z nájdených zariadení je zistené, či disponuje patričnou službou (v našom prípade službou hry). Pokiaľ nie, je nutné nájsť iné zariadenie, pokiaľ áno, je klientovi zaslaná informácia o mape a počte hráčov.

V prípade že dôjde k prerušeniu vyhľadávania zariadenia, služby, alebo sa nepodarí otvoriť spojenie, program sa pokúsi znova inicializovať Bluetooth stack.

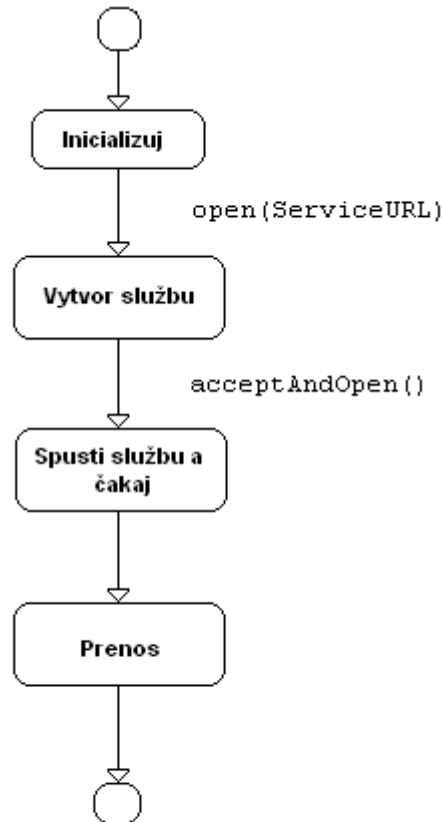


Obrázok 10: Beh serveru

## 6.6.2 Beh klienta

Obrázok č. 11 zobrazuje beh programu klienta. Tu je tiež podmienkou, že zariadenie, na ktorom beží aplikácia, má zapnutú jednotku Bluetooth a že na vzdialenom zariadení PC je zapnutá aplikácia serveru.

Klient hneď pri spustení čaká na pripojenie servera a je pripravený prijímať dáta. Potom, ako sa klient pripojí a hra môže začať.



Obrázok 11: Beh klienta

### 6.6.3 Použitie J2ME na prepojenie telefónov cez Bluetooth

Prepojenie dvoch zariadení cez Bluetooth je prakticky na aplikácii nezávislá operácia. Ide o spojenie klient multiserver. Server pripojí všetky zariadenia, ktoré podporujú danú službu a pokúsi sa o prenos dát.

Na rozdiel od klasického spojenia Klient-Server, ako ho poznáme u aplikácií využívajúcich protokol TCP/IP, klienti sa nepripájajú k už bežiackej aplikácii typu server. U zariadení využívajúcich na komunikáciu Bluetooth a s ním spojený Bluetooth protokol stack je to presne naopak. Najprv je potrebné zaregistrovať na klientskom zariadení službu a spustiť čakane na host (server) aplikáciu. Následne

spustíme host (server) aplikáciu, ktorá vyhľadá zariadenia v okolí a pokúsi sa pripojiť tie zariadenia, ktoré poskytujú hostom požadovanú službu. Až potom môže prebiehať samotná komunikácia.

Na prepojenie zariadení pomocou technológie Bluetooth je možné použiť jeden z protokolov OBEX, RFCOMM, L2CAP v poradí od najvyššieho po najnižší (a to tak, že najvyšší je odvodený od bezprostredného následníka). V programe bol použitý práve protokol RFCOMM simulujúci sériové spojenie pre jeho podporu prúdovej (angl.: stream) komunikácie.

## Prepojenie klienta

Na začiatku je nutné získať objekt triedy `DiscoveryAgent`, na základe nášho (lokálneho) zariadenia. Ten umožňuje nastaviť, akou metódou má byť zariadenie vyhľadané. V tomto prípade ide o konkrétnu službu (hra) preto je vhodnejšia metóda `LIAC`.

Ďalšou časťou je registrácia služby. Najprv je nutné nastaviť unikátny identifikátor `UUID` (48 bit), ktorý bude jednoznačne určovať službu. Ten je nevyhnutnou súčasťou `URL` nutnej na registráciu služby. Zavolaním metódy `(StreamConnection)((StreamConnectionNotifier) Connector.open(url)).acceptAndOpen()` dá klient okolitým zariadeniam, ktoré majú záujem o jeho službu, že je pripravený ku komunikácii. Metóda `acceptAndOpen()` spôsobí pozastavenie vlákna, z ktorého bola volaná. Po pripojení klienta hostom (serverom) je vlákno odblokované a vrátená trieda prúdového spojenia. Samotná komunikácia potom prebieha pomocou vstupno-výstupných prúdov.

## Prepojenie Serveru

Prvým krokom pri prepožovaní host (server) aplikácie je zamedzenie vyhľadateľnosti zariadenia pomocou metódy `local_device.setDiscoverable(DiscoveryAgent.NOT_DISCOVERABLE)`.

Pomocou metódy `startInquiry()` triedy `DiscoveryAgent` sa spustí vyhľadávanie Bluetooth zariadení, ktoré sú v dosahu. Tá potom nájdené zariadenia pomocou predáva objektu triedy `InquiryListener`.

Následne je pre všetky nájdené zariadenia spustené prehľadávanie služieb, ktoré obsahujú požadovaný identifikátor UUID. To sa vykoná pomocou metódy `disc_agent.searchServices()`. Nájdené služby sú potom uchované pomocou objektu triedy `ServiceListener`, podobne ako to bolo v prípade zariadení u triedy `InquiryListener`.

Všetky zariadenia poskytujúce danú službu (game) môžeme nájsť vo vektore `FoundServiceRecord` triedy `ServiceListeners`.

Tento zoznam bude zobrazený užívateľovi a on môže vybrať hráčov z tohto zoznamu (vybrať jedno alebo viac zariadení na pripojenie). Alebo inak povedané, odstrániť všetky zariadenia `FoundServiceRecords` zoznamu, ku ktorým sa nechce pripojiť.

Z nájdených zariadení sa vyfiltrovali tie, ktoré nemali takú službu, ako sme hľadali. Pre zvyšné zariadenia môžeme vytvoriť spojenie a otvoriť prúdy pre zápis a čítanie bajtov. URL adresu spojenia dostaneme volaním metódy `((ServiceRecord) serv_listener.FoundServiceRecords.elementAt( i )).getConnectionURL(0, false)`. Pomocou takto získanej adresy už potom jednoducho otvoríme spojenie, podobne ako tomu bolo u Klientskej časti aplikácie.

Triedy `InquiryListener` a `ServiceListener` sú vytvorené implementáciou rozhrania `DiscoveryListener`.

Implementujeme triedu `InquiryListener`, pomocou ktorej získame zoznam dostupných zariadení. Z tohto zoznamu potom môžeme vyfiltrovať zariadenia s našou službou. To sa dosiahne implementovaním metódy `deviceDiscovered()`, ktorá je automaticky volaná vždy pri nájdení nového zariadenia.

Ďalej je potrebné implementovať triedu `ServiceListener`, ktorá bola použitá ako parameter metódy `searchServices` inštancie triedy `disc_agent`.

Inštancia triedy uchováva všetky nájdené služby. Na základe týchto služieb môžeme vyfiltrovať zariadenia zo zoznamu `cached_devices` triedy `InquiryListener`. To dosiahneme implementáciou metódy `servicesDiscovered()`, ktorá je volaná, keď je nájdená nejaká služba. Táto metóda nastaví `FoundServiceRecord` tak, že sa zastaví vyhľadávanie služieb.

## 6.7 Testovanie programu

Testovanie prebiehalo na štandardnom farebnom emulátore od Sun Microsystems. Pri testovaní boli spustené viaceré inštancie tohto emulátoru. Na pokusy boli k dispozícii aj dva telefóny, Sony Ericsson K750i, ktorý podporuje technológiu Bluetooth a nemá operačný systém (ovláda sa pomocou takzvaného firmaware) a Nokia E65, ktorá okrem Bluetooth má aj podporu WiFi a má operačný systém OS Symbian Series 60.

Bluetooth spojenie bolo testované len medzi telefónmi alebo medzi viacerými inštanciami emulátoru. Emulátory totiž nedokázali spoľahlivo spolupracovať s Bluetooth počítača a nešlo naviazať spojenie medzi programom bežiacim na počítači v emulátore a na mobilnom telefóne. Na druhej strane spojenie WiFi emulátoru nerobí žiaden problém a teda nebol problém testovať program medzi počítačom a mobilnými telefónmi.





Obrázok 12: Emulátory spoločnosti Sun pri testovaní

Testovanie vyžadovalo paralelné ovládanie všetkých hráčov zapojených do hry. Keďže každý emulátor beží v inom okne, nie je možné naraz ovládať viac, ako jeden. Preto bolo nutné vytvoriť generátor náhodného pohybu, ktorý paralelné testovanie umožňoval. To sa dosiahlo nahradením metódy ochyťavajúcej stláčanie kláves príkazom `getDirection(random.nextInt(4))`.

Druhý spôsob testovania spočíval v ovládaní hry nainštalovanej na mobilných telefónoch. Toto je ale veľmi nepohodlný spôsob, keďže uloženie a inštalácia mobilných aplikácií na telefón je, oproti spusteniu emulátora na PC, veľmi zdĺhavý postup. Takýmto spôsobom bola otestovaná len hra dvoch hráčov.

# Kapitola 7

## Známe sieťové hry

V tejto kapitole sú uvedené, známejšie komerčné sieťové hry určené pre mobilné telefóny. Zameriava sa na sieťové možnosti takýchto aplikácií a verziu platformy JME, pre ktorú bol vytvorené.

### 7.1 Snake II

Snake II je video hra pre mobilné telefóny od spoločnosti Nokia [17]. Vznikla v dobách, keď štandardom bol čiernobiely plne grafický display a Java pre mobily ešte len začínala a nemala takú silnú podporu. Hra bola naprogramovaná v prvej verzii J2ME, teda pravdepodobne s konfiguráciou CLDC 1.0 a profilom MIDP 1.0. Podporovala hru jedného alebo dvoch hráčov. Multiplayer bol riešený cez infračervené spojenie (IrDA) a to vyžadovalo priamu viditeľnosť a správne nasmerovanie zariadení voči sebe. Hra mala veľmi nízke FPS (Frame per second), čo vôbec nepôsobilo plynule, skôr trhaným dojmom. Tento fakt pravdepodobne umožňoval hre vysporiadať sa s častými výpadkami spojenia. Hrateľnosť hry nebola veľmi dobrá práve pre už spomínané nevýhody danej technológie spojenia.

## 7.2 Tennis Smash

Hra Tennis Smash je real-time hra s podporou multiplayer cez Bluetooth spojenie [21]. Hra je určená pre dvoch hráčov ktorý proti sebe hrajú tenisový zápas prostredníctvom sieťového spojenia. Hra sa snaží čo najvernejšie simulovať skutočný tenisový zápas so všetkými pravidlami. Pohyb hráčov, je plynulý z čoho sa dá usúdiť, že hra má za štandardných podmienok 25 – 30 FPS. Sieťové spojenie je zabezpečené pomocou technológie Bluetooth, ktorá je dostačujúca pre real-time hru dvoch hráčov. Podľa kvality grafického prevedenia, ktoré je na vysokej úrovni, doby vzniknutia hry a podpory Bluetooth sa dá predpokladať, že hra bola naprogramovaná v J2ME s použitím konfigurácie CLDC 1.1 a profilu MIDP 2.0, ktorý oproti staršej verzii pridáva game API.

## 7.3 Hry podporujúce WiFi

Zatiaľ nie je známa, alebo aspoň dobre zdokumentovaná a na internete dostupná sieťová hra pre mobilné telefóny, bežiacia na platforme JME a podporujúca WiFi spojenie. To môže byť spôsobené aj tým, že dostupnosť tejto technológie je zatiaľ veľmi malá. Týka sa to len niektorých mobilných telefónov najvyššej triedy a smartphone-ov, ktoré svojou funkčnosťou dokážu konkurovať PDA zariadeniam. Taktiež JavaME neumožňuje vývojárom vytvoriť ad-hoc WiFi spojenie. Preto tieto dôvody je aj motivácia vývojárov vytvoriť takúto hru oveľa nižšia, keďže zatiaľ nemá možnosť oslovit' širšiu skupinu ľudí. Pri súčasnom tempe vývoja technológií sa ale aj to môže veľmi rýchlo zmeniť. Veď spomínaná technológia Bluetooth bola ešte pred pár rokmi záležitosťou takzvaných manažérskych telefónov a dnes je podporovaná mobilnými telefónmi strednej a z časti aj nižšej triedy.

# Kapitola 8

## Záver

Táto práca sa zaoberala podrobným štúdiom využitia bezdrôtových technológií vo videohrách pre mobilné telefóny. Dôraz bol kladený na pochopenie a preskúmanie dostupných protokolov používaných práve na mobilných telefónoch.

Aplikácie ako MobilBomber pre IrDA boli hneď zavrhnuté pre ich nevhodnosť hrania na telefónoch. Telefóny museli byť totiž na seba priamo namierené a aj malé odchýlky spôsobovali výpadok spojenia. To robilo hry veľmi náročné na ovládanie, alebo až úplne nehrateľné. Táto technológia tiež prestáva byť používaná na najnovších telefónoch a je nahrádzaná technológiami Bluetooth a WiFi. Bola však ponechaná v komunikačnej knižnici pre jej možné obmedzené využitie v jednoduchých ťahových hrách pre dvoch hráčov.

Bluetooth multiplayer hry fungujú dobre a sú ideálne pre strategické ťahové hry ako aj real-time dvojdimenzionálne hry pre dvoch hráčov, ako bola v práci implementovaná hra MobileBomber. Viac hráčov už spôsobovalo veľké časové zdržanie a hra prestávala byť plynulá. S vidinou novej verzie Bluetooth 4.0 sa ale zdá byť tento problém vyriešený a otvára novú možnosť pre tvorbu trojdimenzionálnych „First Person Shooter“ multiplayerových hier.

Implementácia WiFi spojenia sa zdala byť na prvý pohľad najproblematickejšia. Nedostatok informácií na internete a málo telefónov s jej podporou, sťažuje vývoj takýchto aplikácií. V skutočnosti je ale podpora WiFi v týchto telefónoch (väčšinou s OS Symbian) dobre podchytená v samotných operačných systémoch. Programovanie aplikácie sa teda mení na prácu so socketami, ako pri inom type bezdrôtového spojenia (napr. EDGE). Z čoho vyplýva, že

programátor nemá žiadnu moc nad konfiguráciou WiFi a je nútený podriadiť sa operačnému systému mobilného telefónu, ktoré zatiaľ podporujú len pripojenie cez acces point. Nie je teda možné vytvoriť ad-hoc spojenie a hráč je závislý na dostupnosti nejakého voľného dostupného acces point-u.

Z týchto dôvodov sa v súčasnosti javí ako najlepšie riešenie pre vytváranie multiplayer hier pre mobilné zariadenia Bluetooth, ktorý svojou rýchlosťou dostačuje na hranie 2D hier, aj keď pôvodne nebol koncipovaný na takýto typ komunikácie.

Ďalší vývoj bude záležať práve na tom, kedy sa začnú na trhu objavovať telefóny s novou verziou Bluetooth 4.0.

Nové možnosti vývoja prinieslo aj nedávne spojenie firmy vyvíjajúcej operačného systému OS Symbian so spoločnosťou Nokia a jeho následné uvoľnenie, ako open source [23]. Taktiež štandardizácia rozlíšenia displayu na 320 krát 240 pixlov, by uľahčila vývoj grafických aplikácií pre mobilné telefóny.

Na druhej strane je ale otázne, či by sa ďalší vývoj nemal zamerať na nové platformy, ktoré majú v operačných systémoch bezdrôtovú komunikáciu, ako je Bluetooth alebo WiFi, podchytenú. Sú nimi napríklad Android alebo iPhone.

# Literatúra

- [1] *Mobilné trendy v roku2007* [online]. 2007.  
<<http://mobil.sme.sk/c/3215809/Mobilne-trendy-v-roku-2007-revolucia-sa-odklada.html>>
- [2] Bond, P. *Surve: Games trump movie theaters* [online]. 2009.  
< [http://www.filmjournal.com/filmjournal/content\\_display/news-and-features/news/cinemas/e3i82a4bef380199312dfc68b4a54e12c41](http://www.filmjournal.com/filmjournal/content_display/news-and-features/news/cinemas/e3i82a4bef380199312dfc68b4a54e12c41) >
- [3] Mettala, R., *Bluetooth Protocol Architecture* [online]. 1999.  
<[http://www.bluetooth.com/NR/rdonlyres/7F6DEA50-05CC-4A8D-B87B-F5AA02AD78EF/0/Protocol\\_Architecture.pdf](http://www.bluetooth.com/NR/rdonlyres/7F6DEA50-05CC-4A8D-B87B-F5AA02AD78EF/0/Protocol_Architecture.pdf) >
- [4] *DBm* [online]. <<http://en.wikipedia.org/wiki/DBm>>
- [5] *About the Bluetooth SIG* [online]. <<http://www.bluetooth.com/Bluetooth/SIG>>
- [6] *Specification Documents* [online].  
<<http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/>>
- [7] *Bluetooth* [online]<<http://www.bluetooth.com>>
- [8] *Using the Java APIs for Bluetooth Wireless Technology* [online].  
<<http://developers.sun.com/mobility/apis/articles/bluetoothintro/>>
- [9] *Introdustion to Bluetooth Device Access Guide* [online].  
<[http://developer.apple.com/documentation/devicedrivers/Conceptual/Bluetooth/BT\\_Bluetooth\\_Basics/BT\\_Bluetooth\\_Basics.html](http://developer.apple.com/documentation/devicedrivers/Conceptual/Bluetooth/BT_Bluetooth_Basics/BT_Bluetooth_Basics.html)>
- [10] *A brief history of Wi-Fi* [online].  
<[http://www.coe.montana.edu/ee/rwolff/EE580/history\\_of\\_wifi.htm](http://www.coe.montana.edu/ee/rwolff/EE580/history_of_wifi.htm)>
- [11] *Wikipedia Wireless LAN* [online].  
<[http://en.wikipedia.org/wiki/Wireless\\_LAN](http://en.wikipedia.org/wiki/Wireless_LAN)>
- [12] Ferro, E., Potorti, F. *Bluetooth and WiFi wireless protocols: A survey and a comparison* [online]. 2004.  
<<http://dienst.isti.cnr.it/Dienst/Repository/2.0/Body/ercim.cnr.isti/2004-TR-27/pdf?tiposearch=cnr&langver=>>
- [13] *Bluetooth Technology explained* [online].  
<[http://www.trap17.com/index.php/bluetooth-technology-explained\\_t40377.html](http://www.trap17.com/index.php/bluetooth-technology-explained_t40377.html)>

- [14] Team Nokia. *Games over Bluetooth: Recommendations to Game Developers* [online]. 2003.  
 <[http://fivedots.coe.psu.ac.th/Software.coe/J2ME/bluetooth/Nokia%20Articles/Games\\_over\\_B\\_tooth\\_v1\\_0\\_en.pdf](http://fivedots.coe.psu.ac.th/Software.coe/J2ME/bluetooth/Nokia%20Articles/Games_over_B_tooth_v1_0_en.pdf)>
- [15] Peterka, J. *Prednáška Sítě 2. Verze 3.3.* [online]. 2008. Dostupný z <[earchiv.cz](http://eearchiv.cz)>
- [16] *Mobile leaders to unify the Symbian software platform and set the future of mobile free* [online]. 2008. <<http://www.nokia.com/A4136001?newsid=1230416>>
- [17] *Nokia snake multiplayer* [online]. 2001.  
 <<http://everything2.com/title/Nokia%2520snake%2520multiplayer> >
- [18] Fritzek, F., Reichert, F. *Mobile Phone Programming and its Application to Wireless Networking*, Springer 2007 .
- [19] Diviney, G. *An Introduction to Short-Range Wireless Data Communications*. 2003.
- [20] Mitchell, B. *The 802.11 standards* [online].  
 <[http://compnetworking.about.com/cs/wireless80211/a/aa80211standard\\_2.htm](http://compnetworking.about.com/cs/wireless80211/a/aa80211standard_2.htm) >
- [21] *Tenis smash review* [online]. 2005.  
 <<http://www.mobilegamefaqs.com/reviewfull.php?orig=rev&id=382> >
- [22] Wikipedia *Bluetooth* [online]. <<http://en.wikipedia.org/wiki/Bluetooth>>
- [23] Wikipedia *Symbian OS* [online]. <[http://en.wikipedia.org/wiki/Symbian\\_OS](http://en.wikipedia.org/wiki/Symbian_OS)>
- [Herout] Herout, P. *Učebnice jazyka Java*, Kopp 2000

# Príloha A

## Adresárová štruktúra CD

- **Program** - v tomto adresáre sa nachádza projekt vytvorený editorom NetBeans
  - **MobilBomber** - koreňový adresár projektu, obsahuje hru aj sieťový package Connection
    - **build**
      - **compiled** - obsahuje skompilované triedy projektu
      - **preprocessed** - obsahuje preprocesorom spracované zdrojové kódy
      - **preverified**
    - **dist** - obsahuje spustiteľný \*.jar súbor, ktorý môžeme priamo nainštalovať do telefónu a informačný \*.jad súbor
    - **nbprojekt** - obsahuje súbory vygenerované editorom NetBeans
    - **src** - obsahuje zdrojové kódy a všetky zdroje použité v práci



- **WirelessConnectionLibrary** - sieťový package Connection
  - **build** - obsahuje spracované triedy
  - **dist** - obsahuje jar súbor sieťového pripojenia
  - **nbproject** - obsahuje súbory vygenerované editorom NetBeans
  - **src** - obsahuje zdrojové kódy sieťového package Connection
- **Práca**
  - MobilBomber.pdf
  - MobilBomber.doc

# Príloha B

## Testované telefóny

### Sony Ericsson K750i

#### General

|                    |                  |
|--------------------|------------------|
| Form factor        | Candy-bar        |
| Size (W x H x D)   | 46 x 100 x 21 mm |
| Weight             | 99 g             |
| Battery type       | Li-ion           |
| Talk time (mfr)    | 540 min          |
| Standby time (mfr) | 400 hours        |
| GSM frequencies    | 900, 1800, 1900  |
| GPRS               | Yes              |
| 3G                 | No               |

#### Display

|                     |                           |
|---------------------|---------------------------|
| Main screen (W x H) | 28 x 36 mm                |
| Resolution          | 176x220 pixels            |
| Colour depth        | 18 bits (262,144 colours) |

#### Memory

|                 |       |
|-----------------|-------|
| On-board memory | 34 MB |
|-----------------|-------|

# Nokia E65

## General

Form factor Slider

Size (W x H x D) 49 x 105 x 16 mm

Weight 115 g

Battery type Li-ion

Talk time (mfr) 360 min

Standby time (mfr) 264 hours

GSM frequencies 850, 900, 1800, 1900

GPRS Yes

3G Yes

## Display

Resolution 240x320 pixels

Colour depth 24 bits (16,777,216 colours)

## Memory

On-board memory 50 MB

Memory card type MicroSD

Memory expansion Yes