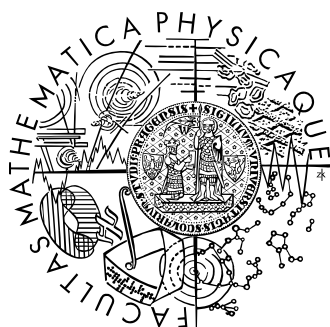


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta
BAKALÁŘSKÁ PRÁCE



Marek Štalmašek

MobilBomber

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Zuzana Vlčková

Studijní program: Informatika, Obecná Informatika

2008

V prvom rade by som chcel poďakovať svojmu vedúcemu Mgr. Zuzane Vlčkovej, za rady a poznámky, ktoré mi poskytla počas celého priebehu písania Bakalárskej práce a s ňou spojeným programom. Taktiež by som chcel poďakovať rodičom a celej rodine. Menovite otcovi Mariánovi za rady a pomoc pri zháňaní materiálov spojených s touto prácou, mame Bele za vytvorenie pracovných podmienok, sestre Michaele za štylistické rady a sesternici Martine Botekovej za sprostredkovanie odbornej pomoci.

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

1.1. Motivácia

1.2. Štruktúra práce

2. Technológie

2.1. Technológia Bluetooth

2.2. Technológia Wifi

2.3. Porovnanie bezdrôtových technológií

3. Užívateľská dokumentácia

3.1. Inštalácia programu

3.2. Ovládanie a pravidlá hry

4. Programátorská dokumentácia

4.1. MobilBomber

4.2. Preklad a zostavenie programu

4.3. Prepojenie pomocou Wifi

4.4. Prepojenie pomocou Bluetooth

4.5. Testovanie programu

5. Záver

Literatúra

Príloha A

Príloha B

Názov práce: MobilBomber

Autor: Marek Štalmašek

Katedra (ústav): Katedra softwarového inženýrství

Vedúci bakalárskej práce: Mgr. Zuzana Vlčková

e-mail vedúceho: Zuzana.Vlckova@mff.cuni.cz

Abstrakt: Cílem této práce je osvojit si práci s komunikačními protokoly, napsat hru MobilBomber, která dovoluje připojení přes BlueTooth, InfraRed a porovnat tyto technologie mezi sebou a s rozvíjející se WI-FI - určit jejich výhody a nevýhody vzhledem k ostatním komunikačním protokolům.

Hra je určena pro více hráčů a je to varianta síťové hry BomberMan. Důraz bude kladen na zvládnutí práce s komunikačními protokoly.

Názov práce: MobilBomber

Author: Marek Štalmašek

Department Katedra softwarového inženýrství

Supervisor: Mgr. Zuzana Vlčková

Supervisor's e-mail: Zuzana.Vlckova@mff.cuni.cz

Abstrakt: The goal of this work is to master the usage of communication protocols, to develop a game MobilBomber, allowing communication through BlueTooth or InfraRed and to compare these technologies to each other and to WI-FI – describe their advantages and disadvantages.

The game will be in multiplayer mode with more players. The emphasis is laid on the mastering the communication through specified protocols.

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, 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 Štruktúra práce

Kapitola 2 sa 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 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 4 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.

V Kapitola 5 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

Bezdrôtová technológia Bluetooth [3] je rádiový štandard určený pre spájanie zariadení na krátku vzdialenosť, ktorého pôvodnou úlohou bolo odbremeniť elektronické zariadenia od káblov, ktorými sú medzi sebou prepojené.

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

2.1.1 Kľúčové vlastnosti

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

Tabuľka 1 uvádza jednotlivé triedy Bluetooth a ich maximálny možný dosah. Hodnoty sa líši podľa prostredia, kde Bluetooth prenos prebieha (budovy, voľné priestranstvá, atd.). Z tabuľky 1 je zrejmé, že dosah súvisí s vysielacím výkonom zariadenia. Mobilné telefóny sa nachádzajú 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 1: Dosah technológie Bluetooth [4]

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 webových stránkach [7].

2.1.3 Využitie technológie Bluetooth

Technológia Bluetooth má v dnešnej dobe nespočetne veľa rôznych využití v mnohých odvetviach. Nasledujúci prehľad uvádza niektoré oblasti, v ktorých sa Bluetooth technológia už uplatňuje.

- Dátová komunikácia medzi dvoma rovnakými zariadeniami (mobilné telefóny, PDA zariadenia, notebooky, atď.)
- Dátová komunikácia medzi dvoma rôznymi zariadeniami (PC a mobilný telefón, PC a PDA, mobilný telefón a tlačiareň, atď.)
- Prenos hlasu medzi mobilným telefónom a handsfree sadou
- Prepojenie periférnych zariadení k PC (klávesnice, myš, slúchadlá, tlačiareň)
- Ad-hoc lokálne siete medzi počítačmi
- Diaľkové ovládanie zariadení v domácnosti (PC, atď.)

Ďalšie využitie Bluetooth môže špecifikovať sám programátor. A to jednoducho tak, že napíše aplikáciu pracujúcu s týmto rozhraním a nainštaluje ju vhodne vybaveného zariadenia. Prípadne si môže tiež navrhnuť samotné zariadenie tak, aby presne vyhovovalo požiadavkám jeho aplikácie.

2.1.4 Bluetooth protocol stack

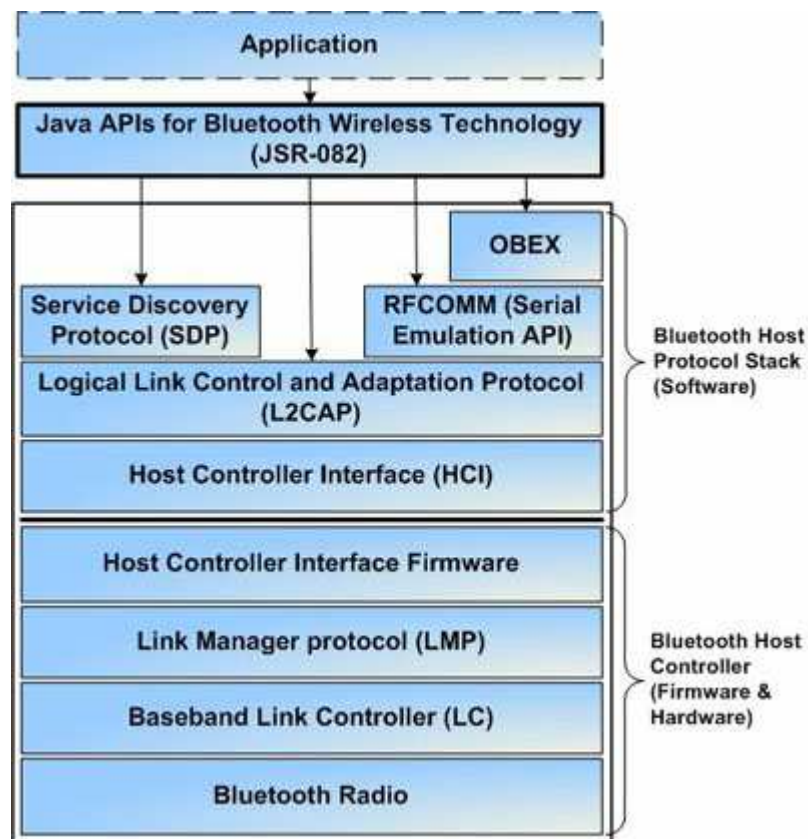
Jadrom špecifikácie Bluetooth je tzv. „Bluetooth protocol stack“, ktorý definuje, ako celá technológia funguje. Rozdeľuje Bluetooth na jednotlivé vrstvy, podobne ako napríklad ISO model OSI u počítačových sietí. Jeho schému zobrazuje Obrázok 1, pričom nižšie vrstvy od tých vyšších oddeľuje tzv. HCI (Host Controller Interface). Najnižšie sa nachádza Radio vrstva, ktorá sa stará o moduláciu a demoduláciu signálu a popisuje fyzické požiadavky na Bluetooth vysielač 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 udržuje 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), hodiaca sa hlavne pre dátovú komunikáciu.



Obrázok 1: Bluetooth protocol stack [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, atd. 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.

2.1.5 Profily Bluetooth

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:

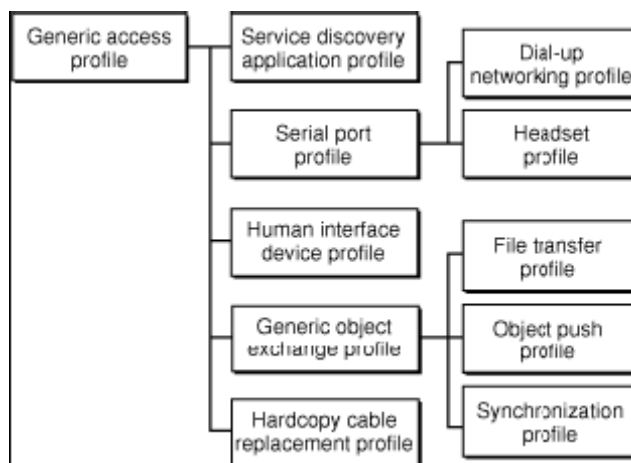
- Závislosti na iných profiloch

- Navrhované formáty užívateľského rozhrania

- Časti Bluetooth protocol stacku využívané týmto profilom

Charakteristika niektorých profilov [9] je uvedená v nasledujúcom prehľade, pričom ich

hierarchiu zobrazuje Obrázok 2.



Obrázok 2: Hierarchia základných Bluetooth profilov [9]

Hlavným (base) profilom je GAP (Generic Access 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 zapôsob komunikácie Bluetooth headset sady s počítačom či mobilným telefónom. Headset po prepojení funguje ako vstupne/výstupné vzdialene 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

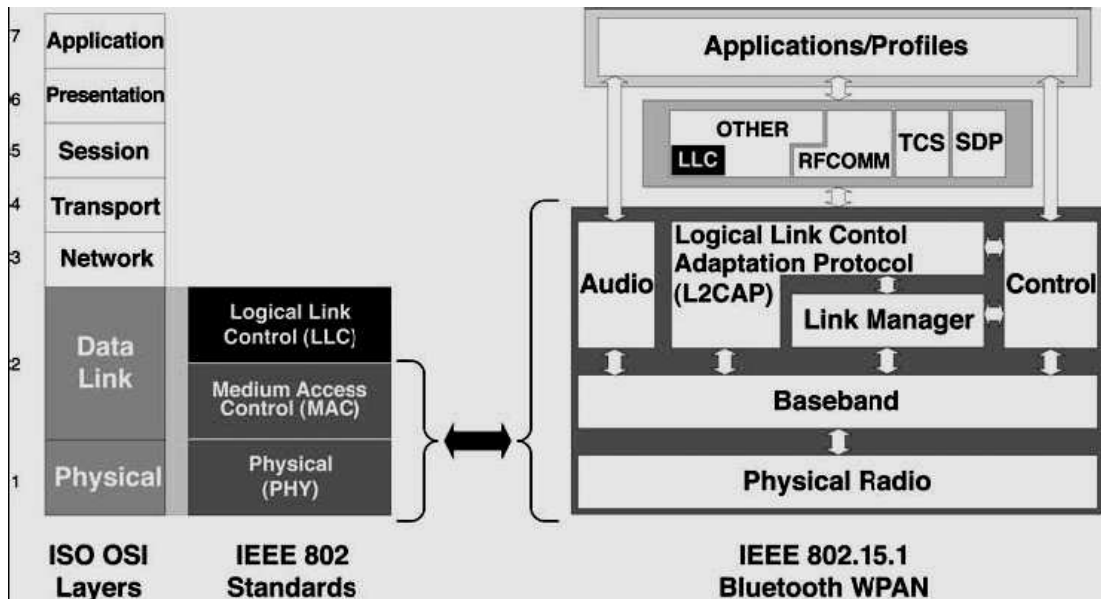
dajú pripojiť tlačiarne, notebooky a PDA. Na trhu sa dokonca objavujú prvé mobilné telefóny podporujúce túto technológiu bezdrôtovej komunikácie.

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 [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) [14].

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 Porovnanie bezdrôtových technológií

Tabuľka 2 uvádza porovnanie troch najbežnejších bezdrôtových technológií, s ktorými sa dnes môžeme stretnúť. Infraport (IrDA) sa používa na prepojenie dvoch zariadení (PC a PDA, PC a mobil, mobil a mobil, atd.) na krátkou vzdialenosť a priamu viditeľnosť. Prenosové rýchlosti sú u tohto typu prepojenia veľmi nízke. 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.

Technológia Bluetooth, jedna z technológií, má dosah rádovo niekoľko metrov, nízky vysielač výkon a teda malé nároky na napájací prúd. Do budúcnosti sa javí ako veľmi perspektívna v mnohých oblastiach elektroniky.

Feature & Function	Infrared	802.11 Wireless LANs	HomeRF	Bluetooth™
Connection Type	Infrared, narrow beam	Spread spectrum	Spread spectrum	Spread spectrum
Spectrum	Optical 850 nm	RF 2.4 GHz	RF 2.4 GHz	RF 2.4 GHz
Transmission Power	100mW	100mW	100mW	1mW
Data Rate	16Mbps	1Mbps, 2Mbps	1Mbps, 2Mbps	1Mbps
Range	1 meter	100 meters	Typical home	3 meters
Supported Devices	2		127	8
Voice Channels	1	VOIP	6	3
Addressing	32 bit physical ID	48 bit MAC	48 bit MAC	48 bit MAC

Tabuľka 2: Porovnanie bezdrôtových technológií [13]

Napriek tomu že bol Bluetooth pôvodne navrhnutý kvôli odstráneniu káblov spájajúcich periférie, došlo k jeho veľkému rozvoju aj v iných oblastiach. Dnes už nie je nič neobvyklého na tom, keď si dvaja ľudia vymieňajú cez Bluetooth aplikácie alebo mediálny obsah, alebo keď si idete vytlačiť fotky tak, že mobil spojíte s tlačiarňou cez Bluetooth a stlačením jednej jedinej klávesy tlačíte dokument. Bluetooth je veľmi zaujímavá technológia, ktorá má určite budúcnosť.

	IR	Bluetooth	Wireless
Range (meters)	1	10	100
Rate (Mbps)	1	1	3
Networked	No	Yes	Yes
Price/Node	\$5	\$10 Target	\$400
Status	100 Million	Emerging	Here Now
Security	No	Level 2 Encryption	Level 2 Encryption
Standards	De-Facto	Specifications	Several

Obrázok 4: Porovnanie technológií

Dôležitou otázkou pri vytváraní sieťových hier je aj cenová relácia pripojenia. Nie každý užívateľ má dosť peňazí na to aby si mohol zaobstarat' prístupový bod (známy je skôr anglický názov Access point) na Wifi pripojenia. A aj použitie technológií v samotných telefónoch predražuje tieto zariadenia.

Obrázok 4 približne zobrazuje cenové náklady na jednotlivé bezdrôtové spojenia.

Kapitola 3

Užívateľská dokumentácia

3.1 Inštalácia programu

Inštalácia MIDletu, tak sa volá mobilná aplikácia resp. program, vôbec nie je náročná. Človek nemusí byť programátorom, aby spojznil 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_suboru.jar*. Spolu s týmto súborom býva distribuovaný aj súbor *nazov_suboru.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 použite návod dodaný k telefónu

3.1 Ovládanie a pravidlá hry

Po inštalácii Java aplikácie do mobilu je štandardne aplikácia zaradená do zoznamu alebo priečinku inštalovaných aplikácií. Pre spustenie aplikácie je nutné ho nájsť. Väčšinou je dostupný priamo v hlavnom menu telefónu. Spustenie aplikácie prevedieme kliknutím hlavného tlačila joysticku. V prípade, že telefón nereaguje, nájdete bližšie informácie v návode dodanom k mobilnému telefónu.

Po spustení aplikácie zvolíme typ pripojenia a pozíciu telefónu v rámci hierarchie siete. 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 AccesPoint-u, telefón nás upozorní, že aplikácia chce naviazať bezdátové spojenie cez rozhranie WiFi. 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.

Hráč sa ovláda nasledovnými klávesmi mobilného telefónu:

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

Bomby:

Hráč nesie so sebou jednu až osem bômb. Tieto môže položiť bezprostredne za sebou. 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 nastraženiu 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 krát 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 4

Programátorská dokumentácia

4.1 MobilBomber

V tejto kapitole je rozobraná štruktúra programu, niektoré algoritmy a dátové štruktúry. Nachádza sa tu aj postup zostavenia a prekladu zdrojových kódov.

4.1.1 Cyklus hry

Cyklus hry, anglicky tiež game engine, je hlavnou časťou drvivej väčšiny počítačových hier. Nachádza sa v ňom všetko čo je nevyhnutné pre beh hry a jej vykreslenie na obrazovku.

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.

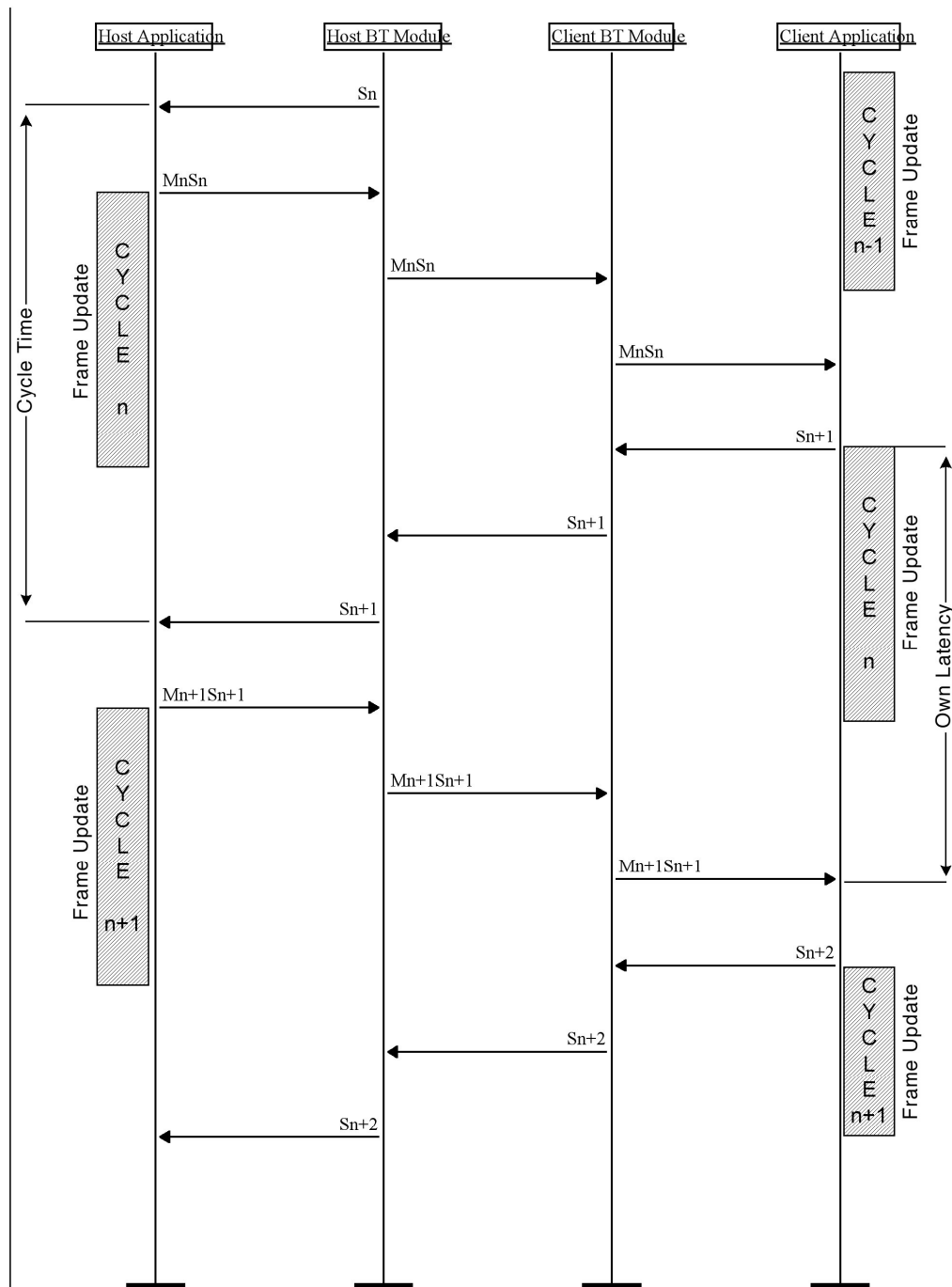
4.1.2 Aktualizácia stavu hry

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 ide o spojenie).

Herný cyklus serveru začína prijatím informácií od klienta. Následne server vyhodnotí svoj herný stav a spolu s informáciami o stave klienta, ktoré od neho obdržal, ich posielajú späť klientovi. 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 servere aj klientovi. Naopak pri zasielaní informácií od klienta, vlákno Bluetooth spojenia posielajú 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.



Obrázok 5: Porovnanie technológií [14]

4.1.3 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. Triedu 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-buffra. 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.

4.2 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. Ďalšia možnosť ako preložiť zdrojové kódy, je použiť príkazovú riadku. Na to slúži príkaz `javac [mena suborov]`.

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.

4.3 Prepojenie pomocou WiFi

Prepojenie mobilných telefónov cez Wifi pracuje na základe sieťového protokolu TCP/IP. V Jave 2 Micro Edition existuje v profile MIDP 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.

```
//prerobit
ServerSocketConnection server = (ServerSocketConnection)
Connector.open("socket://:8888");
```

Po vytvorení societu chceme naviazať spojenie s potenciálnymi klientmi. Metóda *acceptAndOpen()* aktivuje server do vyčkávacieho režimu. To spôsobí zastavenie vlákna Serveru podobné prerušeniu pri čakaní na vstup[Herout]. K znovuoživeniu 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 `openDataOutputStream` otvoríme výstupný prúd pre zápis do socketu. A pomocou `openDataInputStream` zase otvoríme vstupný prúd pre čítanie zo socketu.

```
//otvorenie streamov
DataOutputStream is = client.openDataOutputStream();
DataInputStream os = client.openDataInputStream();
```

Na zápis je najvhodnejšia metóda `writeUTF`. Ktorá zachováva diakritiku a na telefónoch je najširšie podporovaná. Pre načítanie je potom analogicky určená metóda `readUTF`.

```
//samotne citanie a zápis pomocou prúdov
String s;
```

```
s = is.readUTF();
```

```
os.writeUTF("ahoj");
```

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();
```

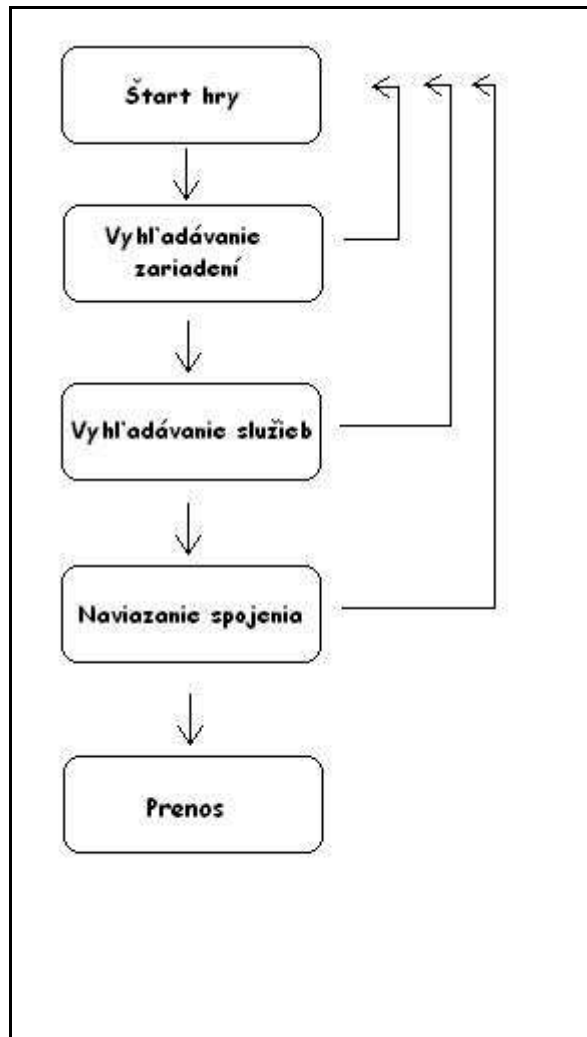
4.4 Prepojenie pomocou Bluetooth

Obrázky 6 a 7 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. Akonáhle 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.

4.4.1 Beh serveru

Obrázok c. 6 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.

Server hneď pri spustení čaká na pripojenie klienta a je pripravený prijímať dáta. Potom, ako sa klient pripojí a požiada o informácie o mape a počte hráčov, môže hra začať.

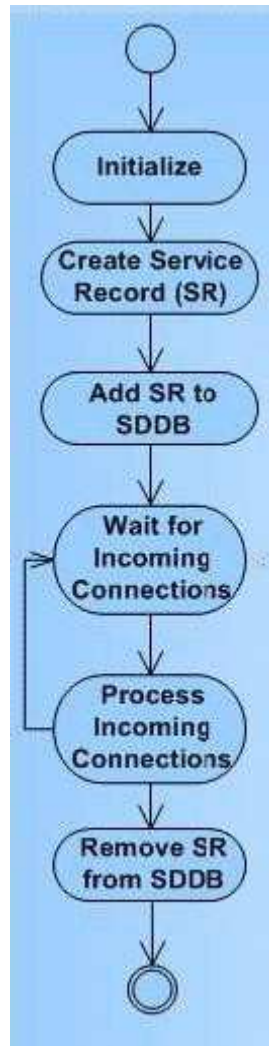


Obrázok 6: Beh serveru

4.4.2 Beh klienta

Obrázok c. 7 zobrazuje beh programu klienta. Tu je tiež podmienkou, že zariadenie, na ktorom beží aplikácia, má zapnutú jednotku Bluetooth a že na vzdialenom PC je zapnutá aplikácia serveru.

Pri štarte klient hľadá všetky dostupné zariadenia, po výbere niektorého z nájdených zariadení je zistené, či disponuje patričnou službou. Pokiaľ nie, je nutné nájsť iné zariadenie, pokiaľ áno, je klientovi zaslaná informácia o mape a počte hráčov.



Obrázok 7: Beh klienta

4.4.3 Použitie J2ME na prepojenie telefónov cez Bluetooth

Prepojenie dvoch zariadení cez BT je prakticky na aplikácii nezávislá operácia. Jednu z možných prepojení uvádza Nokia [14]. Ide o spojenie klient multiserver. Server pripojí všetky zariadenia ktoré podporujú danú službu a pokúsi sa o prenos dát.

Prepojenie klienta

Na začiatku je nutné získať takzvaného „Discovery agenta“, na základe nášho (lokálneho) zariadenia. Ten slúži na vyhľadanie dostupných zariadení

```
// Ziskanie objektu local device
LocalDevice local_device = LocalDevice.getLocalDevice();
// Ziskanie objektu discovery agent
DiscoveryAgent disc_agent = local_device.getDiscoveryAgent();
// Nastavenie zariadenia do módu limitovaného prístupu. Patraci sken
// bude nasluchat len LIAC.
local_device.setDiscoverable( DiscoveryAgent.LIAC );

// Rob prehľadávanie služieb na vsetkych najdenych zariadeniach
String service_UUID = "0000000000010008000006057028C19";
// Ziskaj meno hraca
// Toto moze byt meno ktore hrac zmenil a ulozil
// non-volatile memory. Ako vychodzie (ked je hra spustena prvý
// krát)
// by malo byt nastavene local friendly name.
String player_name = local_device.getFriendlyName();
// Otvor spojenie, pozn: meno je atribut ID 0x0100
String url = "btspp://localhost:" + service_UUID + ";name=" +
player_name ;
StreamConnectionNotifier notifier = (StreamConnectionNotifier)
Connector.open( url );
// Pockaj na niekoho pre spojenie (pozn: mozete prerusit toto
// cakanie
// iba ak je zavolane notifier.close() z ineho vlakna.
// Toto je dolezite ak chcete ponuknut uzivatelovi UI (User
// Inatinterface) zrusit zalozenie spojenia.)

StreamConnection con = (StreamConnection) notifier.acceptAndOpen();
// otvori input stream
InputStream is = con.openInputStream();
// otvori output stream
OutputStream os = con.openOutputStream();
// Zariadenia su teraz spojene:
// Spustite hru / vymente data ...
// Multiplayer spojenie je prerusene: Odpoj zariadenia
// zavri input stream
is.close();
```

```

// zavri output stream
os.close();

// Zavri spojenie
con.close();

```

Prepojenie Serveru

```

// Ziskaj object local device
LocalDevice local_device = LocalDevice.getLocalDevice();
// Ziskaj objekt discovery agent
DiscoveryAgent disc_agent = local_device.getDiscoveryAgent();
// Zamedzenie skenovani page scan a inquiry scan
local_device.setDiscoverable( DiscoveryAgent.NOT_DISCOVERABLE );

// vytvor object inquiry listener vyhľadanie zariadenii
InquiryListener inq_listener = new InquiryListener();
synchronized( inq_listener )
{
    // zacni dotazovanie limitovanim spojenim and instaluj inquiry
    //listener
    disc_agent.startInquiry( DiscoveryAgent.LIAC, inq_listener );
    // cakaj
    inq_listener.wait();
}

// Vyhľadaj danu sluzbu na vsetkych zariadeniach
UUID[] u = new UUID[1];
// Note: don't use this UUID in your own MIDlets.
// You have to create an own UUID for each MIDlet that you write:
u[0] = new UUID( "00000000000010008000006057028C19", false );
int attrbs[] =
{ 0x0100 };

// Ziskany zaznam o sluzbe by mal obsahovat meno hraca
// (service name)
Enumeration devices = inq_listener.cached_devices.elements();
ServiceListener serv_listener = new ServiceListener();
while( devices.hasMoreElements() )
{

```

```

synchronized( serv_listener )
{
    // na danom zariadeni hladaj sluzbu
    disc_agent.searchServices( attrbs, u, (RemoteDevice)
    devices.nextElement(), serv_listener );
    // cakaaj
    serv_listener.wait();
}
}

```

Všetky zariadenia poskytujúce danú službu (game) môžeme nájsť v `serv_listener.FoundServiceRecord` zozname.

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.

```

//Tu vypiseme mena vzdialenych hráčov.
int number_of_players = serv_listener.FoundServiceRecords.size();
String player_name;
for( int i = 0; i < number_of_players; i++ )
{
    // Ziskaj meno ktore sa nachadza ako servis name
    // (0x0100) v zazname sluzieb
    player_name = (String) ((ServiceRecord)serv_listener.
    FoundServiceRecords.elementAt(i)).getAttributeValue(
    0x0100 ).getValue();
    // vypis meno
    System.out.println( player_name );
}

```

Z nájdených zariadení sme vyfiltrovali tie, ktoré nemali takú službu, ako sme hľadali. Pre vyšné zariadenia môžeme vytvoriť spojenie a otvoriť prúdy pre zápis a čítanie bajtov. Po skončení vstupno-výstupných operácií ich zase uzavrieme.

```

// Po vyfiltrovaní, ostavajuce zariadenia budu pripojene
String url;
StreamConnection con[] = new StreamConnection[number_of_players];
InputStream is[] = new InputStream[number_of_players];
OutputStream os[] = new OutputStream[number_of_players];
for( int i = 0; i < number_of_players; i++ )

```

```

{
    // Ziskaj url z daneho zariadenia/sluzby
    url = ((ServiceRecord) serv_listener.FoundServiceRecords.
        elementAt( i )).getConnectionURL( 0, false );
    // Otvor connection
    con[i] = (StreamConnection) Connector.open( url );
    // Otvor input stream
    is[i] = con[i].openInputStream();
    // Otvor output stream
    os[i] = con[i].openOutputStream();
}
// Teraz su zariadenia spojene:
// Spusti hru / vymen data
// Spojenie prerusene: Odpoj zariadenia
for( int i = 0; i < number_of_players; i++ )
{
    // Zatvor input stream
    is[i].close();
    // Zatvore output stream
    os[i].close();
    // Zatvor connection
    con[i].close();
}

```

Implementujeme triedu `InquiryListener`, pomocou ktorej získame zoznam dostupných zariadení. Z tohto zoznamu potom môžeme vyfiltrovať zariadenia s našou službou.

```

// Pre kazde najdene zariadenie je zavolana metoda deviceDiscovered
// Ak je priskum ukonceny (okolo 10 sekund) je zavolana metoda
//inquiryComplete

```

```

private class InquiryListener
implements DiscoveryListener

{
    // Zoznam najdenych zariadeni
    public Vector cached_devices;
    /**
    * Constructor
    */
    public InquiryListener()
    {

```

```

cached_devices = new Vector();
}

/**
 * Zavolana, ked je najdene zariadenie pocas priskumu. Patrania
 * po zariadeniach, ktore su dostupne. Rovnake zariadenie
 * moze byt vratene viac krat.
 */
public void deviceDiscovered( RemoteDevice btDevice,
    DeviceClass cod )
{
    // Filter CoD: uklada zariadenie len v pripade ze ide o mobilny
    // telefom.
    // Pametej ze kod v koncepte SDK je nespravby.
    int major = cod.getMajorDeviceClass();
    if( major == 0x0200 )
    { // Je to dalsi telefon, uloz ho
        if( ! cached_devices.contains( btDevice ) )
        { // Ale len ked nieje v zozname
            // (same device might be reported multiple times)
            cached_devices.addElement( btDevice );
        }
    }
}

/**
 * Zavola ked je prieskum skonceny
 */
public void inquiryCompleted( int discType )
{
    // prieskum skonceny
    synchronized( this )
    {
        this.notify();
    }
}

/**
 * Nepouzajte pre prieskum, ale musia byt deklarovane
 */
public void servicesDiscovered( int transID, ServiceRecord[]
servRecord )
{}
public void serviceSearchCompleted( int transID, int respCode )
{}
}

```

Ďalej je potrebne implementovať triedu `ServiceListener`, ktorá bola použitá ako parameter metódy `searchServices` inštancie triedy `disc_agent`. Táto trieda uchováva všetky nájdené služby. Na základe týchto služieb môžeme vyfiltrovať zariadenia zo zoznamu `cached_devices` triedy `InquiryListener`.

```
private class ServiceListener
implements DiscoveryListener
{
    // holds udrzuje najdene sluzby
    private Vector FoundServiceRecords;
    /**
    * Constructor
    */
    public ServiceListener()
    {
        FoundServiceRecords = new Vector();
    }
    /**
    * servicesDiscovered.
    * Is Je volane ked je najdena nejak sluzba.
    * Nastavi FoundServiceRecord tak, ze sa zastavi vyhľadavanie
    * sluzieb
    */
    public void servicesDiscovered( int transID, ServiceRecord[]
servRecord )
    {
        DataElement sn; // for service name
        // Na zariadeni bola najdena sluzba
        // Iba prava je zaujmava (ie. Je tam len jedna, lebo hladame
        // jednu)
        // Pridaj sluzbu do zoznamu:
        FoundServiceRecords.addElement( servRecord[0] );
    }
    /**
    * serviceSearchCompleted.
    * Metoda je volana ked je priravene vyhľadavanie sluzieb
    */
    public void serviceSearchCompleted( int transID, int respCode )
    {
        synchronized( this )
        {
            this.notify();
        }
    }
}
```

```

/**
 * Nepoužite pre vyhľadavanie služieb, ale potrebné deklarovať:
 */
public void deviceDiscovered( RemoteDevice btDevice,
DeviceClass cod )
{}
public void inquiryCompleted( int discType )
{}
}

```

4.4 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.

Kapitola 5

Záver

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

Aplikácie 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.

Bluetooth multiplayer hry fungovali dobre a sú ideálne pre strategické ťahové hry ako aj real-time dvojdimenziálne hry pre dvoch až štyroch hráčov, ako bola v práci implementovaná hra MobileBomber. Viac hráčov už spôsobovalo veľkú časové zdržanie a hra prestávala byť plynulá. S vidinou novej verzie Bluetooth 3.0 sa ale zdá byť tento problém ako vyriešený a otvára to novú možnosť pre tvorbu trojdimenziálnych „First Person Shooter“ multiplayerových hier.

Implementácia WiFi spojenia sa zdala byť na prvý pohľad najproblematickejšia. Nedostatok informácií na internete a málo telefónu s podporou, sťažuje vývoj takýchto aplikácií. V skutočnosti ale podpora WiFi je 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 AdHoc 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 či a kedy bude vydaná nová verzia Bluetooth 3.0 a či sa do novej verzie J2ME MIDP 3.0 dostane aj podpora WiFi a možnosť vytvárania vlastných sietí, ako AdHoc, tak aj Infrastructure.

Nové možnosti môže priniesť aj očakávaná spojenie firmy vyvíjajúcej operačného systém OS Symbian so spoločnosťou Nokia a jeho následným uvoľnením, ako open source [10]. Taktiež štandardizácia rozlíšenia display 320 na 240 pixlov, by uľahčila vývoj grafických aplikácií pre mobilné telefóny.

Literatúra

- [1] *Denník SME* [online]. <<http://mobil.sme.sk/c/3215809/Mobilne-trendy-v-roku-2007-revolucia-sa-odklada.html>>
- [2] *ITnews.sk* [online].
<http://www.itnews.sk/buxus_dev/generate_page.php?page_id=6549>
- [3] <http://www.bluetooth.com/Bluetooth/Technology/Building/Research/Bluetooth_Protocol_Architecture.htm>
- [4] <<http://en.wikipedia.org/wiki/DBm>>
- [5] <<http://www.bluetooth.com/Bluetooth/SIG>>
- [6] *Specification Documents*.
<<http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/>>
- [7] <<http://www.bluetooth.com>>
- [8] <<http://developers.sun.com/mobility/apis/articles/bluetoothintro/>>
- [9] <http://developer.apple.com/documentation/DeviceDrivers/Conceptual/Bluetooth/BT_Bluetooth_Basics/chapter_2_section_4.html>
- [10] <http://www.coe.montana.edu/ee/rwolff/EE580/history_of_wifi.htm>
- [11] <http://en.wikipedia.org/wiki/Wireless_LAN>
- [12] Erina Ferro, Francesco Potort (2004): *Bluetooth and WiFi wireless protocols: A survey and a comparison*.
<<http://dienst.isti.cnr.it/Dienst/Repository/2.0/Body/ercim.cnr.isti/2004-TR-27/pdf?tiposearch=cnr&langver=>>>
- [13] <http://www.trap17.com/index.php/bluetooth-technology-explained_t40377.html>
- [14] Team Nokia (2003): *Games over Bluetooth: Recommendations to Game Developers*. <http://fivedots.coe.psu.ac.th/Software.coe/J2ME/bluetooth/Nokia%20Articles/Games_over_B_tooth_v1_0_en.pdf>
- [14] Jiří Peterka: Prednáška Sítě 2. <eearchiv.cz>
- [15] <http://www.nokia.com/A4136001?newsid=1230416>

Príloha A

Adresárová štruktúra CD

- **Hra** - v tomto adresáre sa nachádza projekt vytvorený editorom NetBeans
 - **MobilBomber** - koreňový adresár projektu
 - **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
- **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