

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ZPRACOVÁNÍ OBRAZU A VIDEO NA MOBILNÍCH TELEFONECH

DIPLOMOVÁ PRÁCE

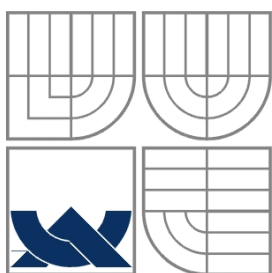
MASTER'S THESIS

AUTOR PRÁCE

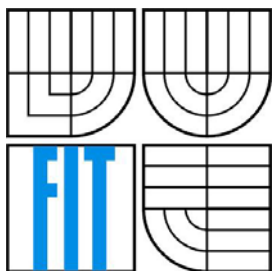
AUTHOR

MARTIN GAZDÍK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

ZPRACOVÁNÍ OBRAZU A VIDEO NA MOBILNÍCH TELEFONECH

IMAGE AND VIDEO PROCESSING ON MOBILE PHONES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN GAZDÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL ŠPANĚL

BRNO 2007

Abstrakt

Tato práce se zabývá problematikou zpracování obrazu a videa na mobilních telefonech s podporou operačního systému Symbian. Popisuje postup při instalaci nástrojů potřebných pro úspěšný vývoj aplikací. Samotnému vývoji však předchází studium a inspirace v existujících aplikacích.

Účelem této práce je vytvořit nástroj pro snadnou práci s integrovaným fotoaparátem a obrázky.

V závěru zhodnotíme dosažené výsledky a možnost budoucího vývoje.

Klíčová slova

Symbian OS, SDK, mobilní telefon, zpracování obrazu, zpracování videa

Abstract

This paper deals with image and video processing on Symbian OS smartphones. Description of required development tools is given, and pros and cons of existing image processing applications are discussed. Afterwards, a new application, fast image viewer and editor, is designed keeping disadvantages of similar applications in mind. Purpose of this work is to make simple and fast tool for easy manipulation with integrated camera and captured images. Results and future development directions are at the end.

Keywords

Symbian OS, SDK, smartphone, image processing, video processing

Citace

Gazdík Martin: Zpracování obrazu a videa na mobilních telefonech. Brno, 2007, diplomová práce, FIT VUT v Brně.

Spracovanie obrazu a videa na mobilných telefónoch

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Ing. Michala Španěla.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Martin Gazdík
21.5.2007

PodĎakovanie

Na tomto mieste by som rád poďakoval Ing. Michalu Španělovi za odbornú pomoc pri spracovaní tohoto projektu.

© Martin Gazdík, 2007.

Táto práca vznikla ako školské dielo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

1	Úvod.....	3
1.1	Smartphone.....	4
1.1.1	Multimediálne a zábavné funkcie	4
1.1.2	Výhody.....	5
1.1.3	Nevýhody.....	5
1.1.4	Siemens SX1	5
2	Teória spracovania obrazu	7
3	Vývoj aplikácií pre OS Symbian	9
3.1	Operačný systém Symbian	9
3.1.1	História.....	9
3.1.2	Verzie	10
3.1.3	Užívateľské rozhrania	10
3.2	Vývojové nástroje	12
3.2.1	Rozdelenie	12
3.2.2	Popis inštalácie.....	14
3.2.3	Odlišnosti pre vývojové nástroje pre Siemens SX1	15
3.3	Vývoj obecných aplikácií.....	16
3.3.1	Vytvorenie novej aplikácie	16
3.3.2	Aplikácia HelloWorld	20
4	Analýza súčasného stavu	29
4.1	Popis a podporované funkcie	29
4.1.1	Aplikácie umožňujúce fotografovanie	29
4.1.2	Prehliadače obrázkov	29
4.1.3	Editory obrázkov.....	30
4.2	Zhrnutie kapitoly	31
5	Návrh aplikácie – inteligentný editor obrázkov	32
5.1	Popis navrhovaných funkcií.....	32
5.1.1	Zachytenie snímku z integrovaného fotoaparátu	32
5.1.2	Zobrazenie náhľadu v jednotlivých adresároch	33
5.1.3	Načítanie a uloženie obrázku z a do súboru.....	34
5.1.4	Rotácia a zväčšenie obrázku	34
5.1.5	Prevod farebných obrázkov na čiernobiele (stupne šedi)	34
5.1.6	Iné obrazové operácie	35
5.2	Zhrnutie kapitoly	35

6	Vlastná realizácia	36
6.1	Štruktúra aplikácie a vývojové nástroje	36
6.1.1	Nastavenie vývojových nástrojov	36
6.1.2	Štruktúra realizovanej aplikácie.....	36
6.2	Vzhľad a popis aplikácie bmpmanip	38
6.2.1	Dostupné funkcie a ovládanie aplikácie.....	38
6.2.2	Funkcia fotografovania a otvárania obrázkov	39
6.2.3	Rotácia obrázkov	40
6.2.4	Uloženie obrázkov	41
6.2.5	Ostatné položky menu.....	41
6.3	Zhrnutie kapitoly	41
7	Výsledky	42
7.1	Analýza problémov	42
7.2	Vyhodnotenie výsledkov	43
8	Záver	45
	Literatúra	47
	Príloha 1.....	48
	Príloha 2.....	53

1 Úvod

Už od začiatku vývoja mobilných telefónov bol kladený dôraz na výslednú veľkosť zariadenia. V súčasnosti, keď užívateľ od svojho telefónu požaduje čoraz viac funkcií, sú výrobcovia neustále nútení svoje produkty zdokonaľovať. Dnešným trendom sa stali tzv. smartphone (inteligentný mobilný telefón), ktorý v sebe kombinuje výhody moderného mobilného telefónu a PDA. Výhodou tohto riešenia je prítomnosť operačného systému. V tejto správe sa zameriame na OS Symbian, ktorý je jedným z najpoužívanejších.

Prvá časť tejto práce je pojatá teoreticky. Obsahuje základný popis, bežne dostupné funkcie mobilného telefónu a informácie o výhodách smartphone a ich koncepcii. Nasledujú základné informácie o spracovaní videa a obrazu. Je tu uvedené, ako prebieha digitalizácia obrazu. Špeciálne sa popisujú funkcie integrovaného fotoaparátu v mobilnom telefóne. Nasledujú základné metódy spracovania obrazu, ktoré sú použité v realizovaných aplikáciách. Na túto časť voľne naväzuje popis štruktúry operačného systému Symbian, jeho postupný vývoj a potrebné informácie pri vývoji aplikácií pre tento systém. Nasleduje podrobný postup inštalácie všetkých potrebných vývojových nástrojov. Ďalej je uvedený príklad vývoja jednoduchej aplikácie pre OS Symbian a postup ktorý je potrebný dodržať pri kompilácii a následnom využití priloženého emulátoru.

Druhá časť práce začína analýzou súčasného stavu dostupných aplikácií pre mobilné telefóny s operačným systémom Symbian. Popísané sú hlavne aplikácie, ktoré využívajú integrovaný fotoaparát alebo nejakým spôsobom pracujú s obrazovými dátami v mobilnom telefóne. Na základe získaných poznatkov je realizovaný návrh plánovanej aplikácie. Požadovanými funkciami je možnosť získania obrázku z integrovaného fotoaparátu a jeho spracovanie. Následne aj práca s obrázkami uloženými v pamäti mobilného telefónu. V závere praktickej časti tejto práce je popísaný postup vývoja navrhutej aplikácie, riešené problémy a popis výslednej aplikácie.

V závere je zhodnotený prínos tejto práce, dosiahnuté výsledky a diskusia možných rozšírení do budúcnosti.

1.1 Smartphone

Často používaný preklad je inteligentný telefón. Takýto telefón obsahuje nejaký otvorený operačný systém (napr. Windows Mobile, Symbian alebo niektorý na báze operačného systému Linux). Vďaka tomu je možné prístroj jednoducho rozšíriť o požadované funkcie (nainštalovaním príslušnej aplikácie). Takéto telefóny (viď. obrázok 1.1) majú bohatú základnú výbavu, ale nie je problém v prípade potreby, doplniť aplikácie potrebné na prácu, komunikáciu a samozrejme aj zábavu.



Obrázok 1.1: Rôzne typy smartphonov. (prevzaté z [2])

1.1.1 Multimediálne a zábavné funkcie

Vďaka neustálemu vývoju a silnej konkurencii výrobcov sa do mobilných telefónov dostávajú funkcie nedávno nepredstaviteľné. Telefón už dávne neslúži len na uskutočnenie telefónneho hovoru.

Majiteľ často chce, aby sa jeho mobilný telefón odlišoval od ostatných, preto je tu možnosť značnej personalizácie. To zahŕňa najmä zmenu pozadia na displeji, zmena zvonenia alebo napr. aj výmena krytov. U najnovších telefónov sa dá dokonca nastaviť ako vyzváňanie akákoľvek skladba vo formáte MP3 alebo WMA.

Posielanie krátkych textových správ je neustále veľmi obľúbené, ale do popredia sa dostávajú správy MMS. Sú to multimediálne správy a môžu obsahovať nielen text, ale aj obrázky, audio a video. Obrázky je možné zhotoviť napríklad integrovaným fotoaparátom. S jeho pomocou je možné natáčať aj krátke videá. Často sa objavujú integrované mp3 prehrávače a v niektorých špičkových modeloch aj integrované rádio.

Ako hlavný zábavný prvok slúži prítomnosť hier. Zo začiatku boli hry veľmi jednoduché, ale s rastúcou obľubou sa neustále zdokonaľujú. Po implementácii technológie JAVA a zavedení farebných displejov sú hry neustále zložitejšie a prepracovanejšie.

S príchodom inteligentných telefónov, ktoré majú implementovaný operačný systém a spravidla väčší displej sa často stretávame s efektívnou možnosťou prehrávania videa a iných doplnkových aplikácií. Vďaka týmto aspektom sa pojem spracovanie obrazu a videa aj v prípade mobilných telefónov stáva zaujímavým pojmom v prípade vývoja nových aplikácií.

1.1.2 Výhody

Medzi základné výhody patrí samozrejme jednoduchá rozšíriteľnosť. Túto vlastnosť klasické mobilné telefóny v podstate nemajú (keď neuvažujeme možnosť JAVA aplikácií). V súčasnosti existuje veľké množstvo kvalitných aplikácií pre Windows Mobile aj Symbian. Mnohé z nich sú dostupné zdarma. Pre ilustráciu uvediem niektoré z nich: internetový prehliadač, emailový klient, multimediálny prehrávač, súborový manažér, autoatlas, atď. Na vysokej úrovni je synchronizácia s počítačom. Ďalšie výhody sú podpora pamäťových kariet, integrovaný fotoaparát, mp3 prehrávač, prehrávač videa, v niektorých modeloch integrované rádio a iné.

1.1.3 Nevýhody

Asi najväčšou nevýhodou je veľkosť takéhoto telefónu. Mnohé typy sa svojou veľkosťou blížia veľkosti vreckových počítačov (PDA). Je to dané hlavne veľkým displejom. Pretože smartphone má spravidla inštalovaný výkonnejší procesor, tak sa výdrž batérií značne znižuje. Pri plnom využití môže táto doba klesnúť až na 1-2 dni. Ďalšou významnou nevýhodou je vyššia cena. Ale po zvážení všetkých vlastností dospejeme k záveru, že je úmerná tomu, čo tieto prístroje dokážu.

1.1.4 Siemens SX1

Pretože k vývoju aplikácií v rámci tejto práce bude používaný mobilný telefón Siemens SX1 (viď. obrázok 1.2), tak na tomto mieste uvediem jeho základné parametre a funkcie (viď. tabuľka 1.1). Tieto informácie môžu napomôcť čitateľovi k získaniu predstavy o schopnostiach tohto telefónu a dať si do súvislostí poznatky z prvej kapitoly.



Obrázok 1.2: Mobilný telefón Siemens SX1 (prevzaté z [3]).

farebný display	65 tisíc farieb
typ farebného displeja	aktívny, TFT
rozmery displeja	35 × 44 mm
rozlíšenie displeja	176 × 220 bodov
systémový konektor pre kábel	áno
infraport	áno
Bluetooth	áno
siete GSM	900, 1 800, 1 900 MHz
užívateľská pamäť	4 MB
pamäťová karta	MMC
štandardný operačný systém	Symbian 6.1 Series 60
multimediálne správy MMS	áno
objekty v odosielanej MMS	text, obrázok, zvuk, video
kalendár	áno
upomienky	áno
synchronizácia	áno
diktafón	2 min
herné technológie	Java v 1.0, Symbian
prehrávanie súborov MP3	áno
FM rádio	stereo
prehrávač médií: video	áno
max. rozlíšenie fotoaparátu	640 × 480 (0,31 Mpx)
možnosť natáčať video	áno
GPRS	trieda 10 (4/2)
EDGE	nie
UMTS	nie
pripojenie k internetu: cez kábel	áno
pripojenie k internetu: cez infraport	áno
pripojenie k internetu: cez Bluetooth	áno

Tabuľka 1.1: Základné parametre mobilného telefónu Siemens SX1.

2 Teória spracovania obrazu

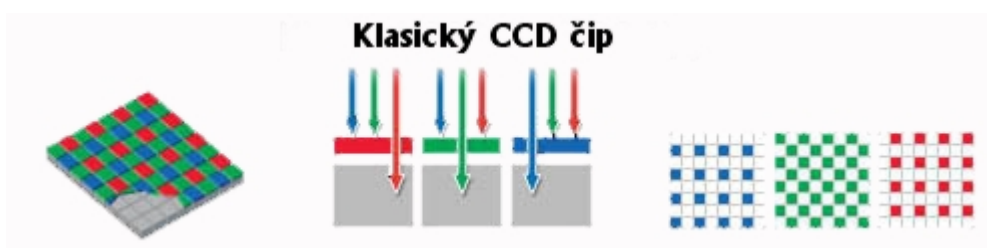
V rámci plánovanej aplikácie, ktorá bude výsledkom diplomovej práce sa plánuje použitie integrovaného fotoaparátu. Preto v krátkosti princíp vzniku obrazu:

V integrovaných fotoaparátoch sa najčastejšie používa snímač typu CCD.

CCD snímače sú malé čipy rôznych rozmerov, zložené z polovodičových elementov, ktoré sú schopné absorbovať svetlo. Po otvorení clony fotoaparátu dôjde na určitú dobu k osvetleniu čipu svetelnými lúčmi. Svetlo si môžeme z fyzikálneho hľadiska predstaviť ako fotóny. Po dopade fotónov na jednotlivé polovodičové elementy (bunky) tohoto prvku získajú elektrický náboj. Na jednotlivé elementy dopadá iné množstvo fotónov. Odtiaľ plyne, že každý element získa iný elektrický náboj - a ten sme schopný zmerať. Ako je teda vidieť, samotný CCD snímač zaznamenáva analógové hodnoty, až obvody za ním urobia sčítanie a digitalizáciu.

K získaniu informácie o farbe sa používa jednoduchý trik. Je známe, že akúkoľvek farbu sme schopní získať zložením troch základných zložiek: červenej, zelenej a modrej. Ich vzájomným pomerným zastúpením potom dostaneme akúkoľvek farbu.

Na CCD snímači to dosiahneme tak, že pred každý element je vložený farebný filter (viď. obrázok 2.1).

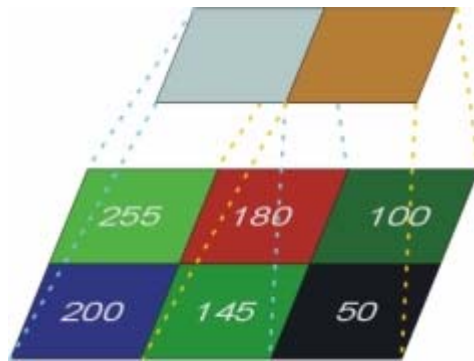


Obrázok 2.1: Klasický CCD čip (prevzaté z [12]).

Z praktických dôvodov sa elementy skladajú do štvoríc, takže jeden z farebných filtrov je zastúpený dvakrát. Voľba daná vlastnosťami ľudského oka, ktoré najlepšie vníma odtiene zelenej farby, padla na dvojicu zelených filtrov. Takže teraz svetlo musí prechádzať cez jednotlivé filtre. Modrý filter prepustí modrú časť spektra a odrazí červenú a zelenú, červený a zelený sa správajú analogicky. Opäť teda dôjde k pohlteniu fotónov a získaniu elektrického náboja, ale teraz jeho veľkosť nezávisí len od intenzity svetla, ale aj od jeho farebného zloženia.

Skutočnú farbu daného pixelu dostaneme matematickou interpoláciou farieb z jednotlivých elementov. Najčastejšie sa používa 24-bitová farebná hĺbka. Tento údaj znamená, že na každú z troch farieb pripadá 8-bitová informácia. To znamená 256 odtieňov danej farby. Ich zložením (interpoláciou) potom dostaneme približne 16 miliónov farebných odtieňov. Spôsob skladania farieb je predvedený na obrázku (viď obrázok 2.2): vezme sa štvorica elementov, ktorá určí farbu jednotlivého

pixelu. V našom prípade dá prvá štvorica hodnoty $R=180$, $G=200$, $B=200$, čo dá dohromady šedú. U druhej štvorice urobíme zloženie analogicky a dostaneme žltohnedú farbu.



Obrázok 2.2: Spôsob skladania farieb nad jednotlivými elementmi CCD čipu (prevzaté z [12]).

V navrhovanej aplikácii, ktorou by mal byť editor obrázkov, predpokladám základné obrazové operácie. Podrobnejší popis je uvedený v kapitole „Návrh aplikácie – editor obrázkov“.

3 Vývoj aplikácií pre OS Symbian

3.1 Operačný systém Symbian

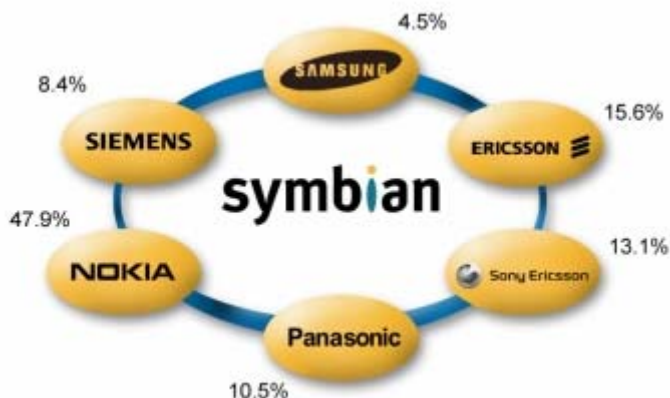
Symbian je momentálne najrozšírenejší operačný systém pre mobilné telefóny. Jeho trhový podiel na poli smartphoneov je viac ako 2/3. To momentálne predstavuje asi 60 rôznych symbianovských telefónov od 8 rôznych spoločností. A ďalšie sú vo vývoji.

3.1.1 História

V júni 1998 sa spojili giganti na trhu s mobilnými zariadeniami – Psion, Nokia, Ericsson a Motorola – s cieľom vytvoriť operačný systém, ktorý by bol dostatočne flexibilný, aby rovnaké jadro mohlo bežať na rôznych typoch prístrojov a vždy s osobitým štýlom vlastným výrobcovi. Založili spoločnosť Symbian Ltd., ktorá sa mala zaoberať vývojom tohto systému a jeho licencovaním. Prvý Symbian vychádzal z jadra systému EPOC, ktorý do spoločnosti priniesol Psion.

O rok neskôr už vyšiel prvý prístroj so Symbianom. Rovnako sa v tomto roku medzi akcionárov dostala aj Matsushita (dnes Panasonic). V roku 2000 sa dostáva na trh prvý symbianový mobilný telefón z dielne Ericssonu. V roku 2001 prišiel prvý mobilný telefón s rozhraním S60 z dielne Nokie - typ 7650 a začal tak éru masívneho nárastu inteligentných telefónov na trhu. Siemens SX1 je odvodený práve od Nokie 7650, preto budeme používať vývojové nástroje z dielne firmy Nokia.

V nasledujúcich rokoch nastali viaceré zmeny vo vlastníctve podielov v spoločnosti a zo skupinky vlastníkov zmizla Motorola spolu so Psionom, naopak pribudol Samsung a Sony Ericsson. Momentálne rozloženie podielov vyjadruje nasledujúci obrázok.



Obrázok 3.1: Rozloženie podielov vo vlastníctve spoločnosti Symbian Ltd. (prevzaté z [1]).

3.1.2 Verzie

Prvá verzia operačného systému vyšla na svetlo sveta pod menom Symbian 6.0. Jej nositeľom bol komunikátor Nokia 9200. Na mierne vylepšenom systéme 6.1 boli postavené známe Siemens SX1, Nokia 7650, 3650 či oba modely N-gage.

V roku 2003 vyšla novšia verzia označená Symbian 7.0, nasledovaná verziou 7.0s, ktorá umožnila podporu sietí 3. generácie. Patrí sem Nokia 6600, 3230 či 6670.

Symbian 8.0 (a neskôr 8.1) vyšiel v roku 2004, okrem niekoľko ďalších zmien prišiel v dvoch verziách. Prvá, využívajúca verziu kernelu EKA1, bola prispôbená na kompatibilitu so staršími verziami zariadení a EKA2 ponúkala niektoré pokrokovejšie funkcie. Na tomto systéme sú založené napríklad Nokia N70, N90 alebo 6630 a 6680.

V minulom roku Symbian uviedol verziu 9.1 (9.0 bola používaná len pre potreby spoločnosti), ktorá priniesla zásadnejšie zmeny hlavne v oblasti ochrany aplikácií a uchovávaného obsahu. Zmeny sú okrem iného zamerané na boj proti škodlivým aplikáciám, ktoré sa v poslednej dobe začínajú vyskytovať čoraz častejšie. Prvým telefónom využívajúcim Symbian 9.1 by mala byť Nokia N91. Pripravované telefóny radu E (E60, E61, E70) takisto od spoločnosti Nokia by mali túto verziu využívať taktiež.

3.1.3 Uživatelské rozhrania

Jedným z hlavných cieľov Symbianu bolo vyrobiť operačný systém dostatočne flexibilný na to, aby prístroje, v ktorých bude nainštalovaný, neboli len obyčajné klony s rovnakým vzhľadom. Preto mohli na rovnakom základe bežať napríklad funkčne vzdialené Nokia 7650 a 9200. Postupom času sa vykryštalizovali 3 hlavné rozhrania:

S60 – najrozšírenejšie a najznámejšie rozhranie (viď. obrázok 3.2). Je určené pre mobilné telefóny ovládané najmä pomocou klasickej klávesnice. Sú to väčšinou mobilné telefóny určené na zábavu, hoci poskytujú aj manažérske funkcie. Vývojom sa zaoberá Nokia a tá je aj najväčším výrobcom telefónov s týmto rozhraním, okrem nej rozhranie S60 využil aj Siemens, Panasonic, atď.

Rovnako ako samotný Symbian, aj jeho rozhrania prešli niekoľkými vývojovými fázami. Prvá edícia S60 je prítomná v telefónoch so systémom Symbian 6.1 a tvorí akýsi základ, na ktorom stavali ďalšie edície. Toto rozhranie má Siemens SX1, Nokia 3650, 7650 či N-gage.

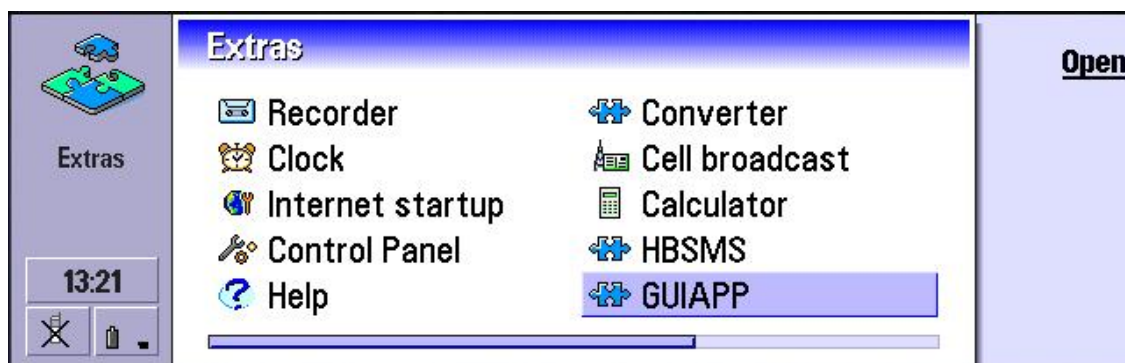
Druhá generácia vyšla v roku 2003 a debutovala v Nokii 6600. Medzi novinky patrila podpora lepšieho rozlíšenia displeja, EDGE, sietí 3G či vylepšený prehliadač webových stránok. Druhá generácia bola použitá na telefónoch so Symbianom verzie 7.x a 8.x, ako napríklad Nokia 3230, 6630, či N70.

Tretia generácia bude nasadená spoločne so systémom Symbian 9.x. Podľa výrobcu bude mať vylepšenú synchronizáciu, kalendár, nový prehrávač hudby s možnosťou sťahovať skladby priamo z internetu, kvalitnejší prehliadač internetu či mnohé vylepšenia pre vývojárov aplikácií.



Obrázok 3.2: Užívateľské rozhranie S60 (prevzaté z [6]).

S80 – rozhranie (viď. obrázok 3.3) sa využíva pre mobilné zariadenia s plnohodnotnou klávesnicou a širokým displejom, napríklad v komunikátoroch 9300 a 9500 od Nokie. Umožňuje pohodlnejšiu tvorbu a upravovanie dokumentov, prácu s multimédiami a ďalšie možnosti uspokojujúce potreby náročných užívateľov.



Obrázok 3.3: Užívateľské rozhranie S80 (prevzaté z [6]).

UIQ – najflexibilnejšie rozhranie (viď. obrázok 3.4) známe hlavne z mobilných telefónov značky Sony Ericsson, použila ho však aj Motorola alebo BenQ. UIQ sa dokáže prispôbiť viacerým zobrazovacím polohám, či už je to klasické ovládanie klávesnicou alebo pomocou dotykového displeja. Rovnako ako S60, prešiel viacerými verziami:

V roku 2000 bola vydaná prvá verzia 1.0 pre mobilné telefóny s operačným systémom Symbian 6.0. O rok neskôr vyšla vylepšená verzia s podporou HSCSD, GPRS či Bluetooth.

V roku 2002 vyšla verzia 2.0, ktorá podporovala Symbian verzie 7.0. hlavným prínosom bola spomínaná flexibilita. Ďalší upgrade na verziu 2.1 sa konal hneď o rok neskôr, odvtedy si však UIQ dalo dlhšiu pauzu, a to až do minulého roku, kedy predstavilo UIQ 3.0.

Nová verzia bola pripravovaná pre nový rad telefónov s operačným systémom Symbian 9.x a okrem krajšieho dizajnu rozhrania bude prínosom hlavne pre operátorov, ktorým sľubuje väčšiu

možnosť prispôbiť mobilný telefón svojim potrebám. UIQ verzie 3.0 bude prítomný aj v očakávanom modeli P990 od spoločnosti Sony Ericsson.



Obrázok 3.4: Užívateľské rozhranie UIQ (prevzaté z [6]).

3.2 Vývojové nástroje

3.2.1 Rozdelenie

Operačný systém Symbian je modulárny a umožňuje výrobcovi rozširovať systém o svoje vlastné funkcie a definovať nové API (application programming interface). Vývojári pomocou nového API môžu pristupovať a pracovať so špecifickým vybavením a funkciami konkrétneho telefónu.

Typickým príkladom grafického užívateľského rozhrania (viď. kapitola 4.1.3), pomocou ktorého sa výrobcovia snažia vzájomne odlišiť a takto zaujať potenciálneho zákazníka. Toto samozrejme prináša svoje úskalia a preto aplikácie programované v C++ nie sú vždy medzi rôznymi telefónmi prenositeľné bez problémov. Vývojár si preto musí zvoliť vhodné SDK a naštudovať rozšírené API, ktoré bude používať vo svojich aplikáciách pre daný typ telefónu.

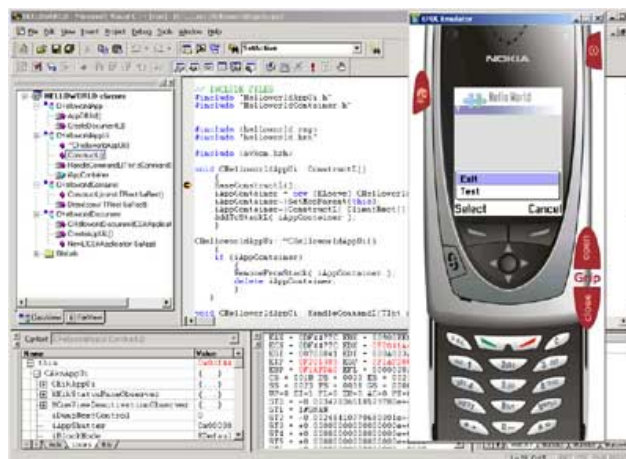
3.2.1.1 SDK – Software Development Kit

Výrobca telefónu poskytuje vývojárom SDK – vývojové prostredie pre daný typ telefónu zdarma alebo po zaregistrovaní.

SDK obsahuje distribúciu štandardných knižníc a API Symbian OS. Toto je doplnené novými knižnicami a API od výrobcu. Súčasťou inštalácie SDK býva dokumentácia, kompilátor na procesorovú platformu telefónu, emulátor, pomocné utility pre správu projektu, vytváranie bitmáp atď.

Sú podporované popredné vývojové nástroje (IDE), ako je napr. MS Visual C++, Borland C++ builder alebo CodeWarrior. Vývojári sú na ne zvyklí a to im umožňuje rýchly vývoj aplikácií.

Výhodou je rozlišovanie farebnej syntaxe, debugging, možnosť pohodlne používať emulátor (viď. obrázok 3.5) alebo vizuálna tvorba resources (definovanie zdrojov).



Obrázok 3.5: Ukážka práce v MS Visual C++ a použitie emulátoru pre Nokiou 7650 (prevzaté z [6]).

3.2.1.2 Prehľad jednotlivých verzií SDK pre S60

Pre verziu OS Symbianu S60 existujú viaceré verzie SDK (viď. tabuľka 3.1). Líšia sa podporovanými verziami mobilných telefónov a vývojovými prostrediami (IDE).

Podporované OS: Windows 2000 SP2, Windows XP SP1/SP2, Windows NT 4.0 SP6

Podporované IDE: Metrowerks Codewarrior, Borland C++ Builder, Microsoft Visual C++ 6, Visual Studio .Net 2003, Carbide.c++

Pre jednotlivé verzie SDK nemusia byť vždy podporované všetky uvedené IDE.

SERIES 60 Developer Platform	Visual Studio or Borland C++ BuilderX	Codewarrior or Carbide.c++	Associated devices
v3	262MB		Nokia N91
DP v2 FP3	150MB		Nokia N70, Nokia N90
DP v2 FP2	127MB	135MB	Nokia 6630, Nokia 6680, Nokia 6681
v2.1	116MB	115MB	Nokia 7610, Nokia 6670, Nokia 6260
v2.0	101MB	106MB	Nokia 6600
v1.2	91MB	115MB	Nokia N-Gage, Nokia 3650, Siemens SX1, Sendo-X

Tabuľka 3.1: Prehľad verzií SDK, veľkosť inšalačného balíčka, príslušné podporované IDE a typy mobilných telefónov, pre ktoré sú jednotlivé verzie primárne určené. (prevzaté z [7]).

Pri verzii SDK v1.2 vo verzii pre MS Visual C++ je nutné poznamenať, že inšalačný balíček neobsahuje inštaláciu produktov Active Perl a Sun JRE. Pre správnu funkciu SDK je nutné tieto dva produkty doinštalovať. Doporučujú sa tieto verzie: Active Perl 5.6.1 alebo novší a Java runtime enviroment (JRE) 1.3.1 alebo novší.

3.2.2 Popis inštalácie

Táto časť obsahuje popis inštalácie SDK a všetkých potrebných nástrojov. Potrebné nastavenia a upozornenie na prípadné úskalía. Najskôr je uvedený všeobecný postup a potom konkrétny pre SDK, ktoré je použité pri vývoji aplikácií pre Siemens SX1.

Všeobecný postup je nasledujúci:

1. Inštalácia zvolenej verzie SDK
2. V prípade potreby nasleduje inštalácia ďalších potrebných nástrojov (napr. Active Perl, Sun JRE, atď.)
3. Inštalácia jednej z podporovaných verzií IDE.
4. Nastavíme potrebné systémové premenné.
5. Doplnenie súborov, ktoré obsahujú šablónu projektu do adresára nainštalovaného IDE, kde sa nachádzajú všetky šablóny. Týmto krokom získa IDE schopnosť založiť nový projekt pre vývoj Symbian OS aplikácie.

Jednotlivé kroky budú zreteľnejšie pri preštudovaní nasledujúceho postupu. Ten popisuje inštaláciu SDK v1.2 pre MS Visual C++ 6.0.

1. Najskôr SDK nainštalujeme do implicitne nastaveného adresára C:\symbian\6.1\Series60.
2. Nainštalujeme Active Perl 5.6.1 a JRE 1.3.1. Inštalujeme do implicitne nastavených adresárov.
3. Inštalácia MS Visual C++ 6.0 do implicitne nastaveného adresára.
4. Skontrolujeme a prípadne nastavíme systémové premenné (viď. obrázok 3.6)

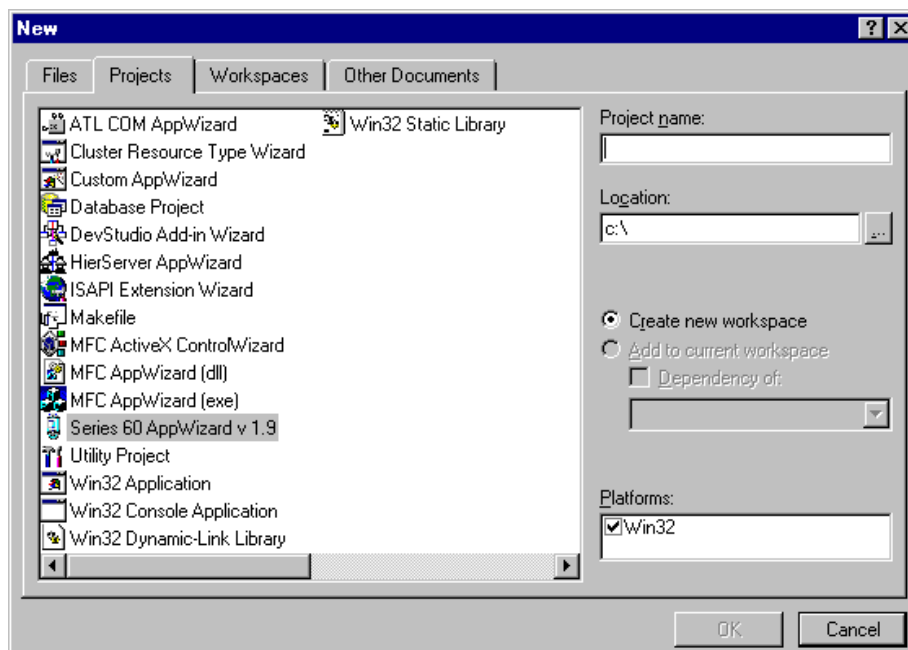
```
PATH=C:\Symbian\6.1\Shared\epoc32\gcc\bin;C:\Symbian\6.1\Shared\epoc32\tools;  
C:\Program Files\Microsoft Visual Studio\VC98\Bin;C:\Perl\bin;  
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;  
C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;  
  
EPOCROOT=\Symbian\6.1\Series60\
```

Obrázok 3.6: Nastavenie systémových premenných (prevzaté z [8]).

Pozor! Cesta definovaná v EPOCROOT nezačína jednotkou! Koreňový adresár zdrojových súborov vyvíjanej aplikácie preto musí existovať na rovnakej diskovej jednotke ako inštalácia SDK.

5. V adresári C:\Symbian\6.1\Series60\Series60Tools\applicationwizard sa nachádzajú dva súbory: avkonappwiz.awx a avkonappwiz.hlp. Tieto súbory nakopírujeme do implicitne nastaveného adresára, v ktorých MS Visual C++ 6.0 uchováva šablóny projektov. V našom prípade je to adresár C:\Program Files\Microsoft Visual Studio\Common\MsDev98\Template.

Ak teraz spustíme MS Visual C++ 6.0 a zvolíme File → New, a v následne zobrazenom okne zvolíme záložku Projects, tak sa v zobrazenom zozname zobrazí položka Series 60 AppWizard v 1.9 (viď obrázok 3.7), pomocou ktorej môžeme založiť nový projekt pre vývoj Symbian aplikácie.



Obrázok 3.7: Okno MS Visual C++ 6.0 pre voľbu nového projektu (prevzaté z [9]).

3.2.3 Odlišnosti pre vývojové nástroje pre Siemens SX1

Po nainštalovaní SDK v 1.2 pre MS Visual C++ je vhodné upraviť vizuálnu podobu vstavaného emulátoru (viď obrázok 3.8). Aplikácie budú vyvíjané pre telefón Siemens SX1, takže je žiadúce, aby mal emulátor rovnaký vzhľad ako originálny telefón (viď obrázok 2.2).



Obrázok 3.8: Pôvodný vzhľad emulátoru Series 60 pre C++ (prevzaté z [9]).

Potrebnú úpravu sa dá dosiahnuť zmenami v adresári C:\Symbian\6.1\Series60\Epoc32\Data. Je nutné aktualizovať súbory epoc.bmp a epoc.ini tak, aby obsahovali podobu a popis telefónu Siemens SX1. Tieto dva súbory nájdeme na priloženom CD v archíve s názvom skin-sx1.zip. Rovnako je možné stiahnuť tento archív aj z internetu. Dostupné napr. na URL:

<http://www.newlc.com/SX1-Skin-for-the-Series60-SDK.html> (máj 2007)

3.3 Vývoj obecných aplikácií

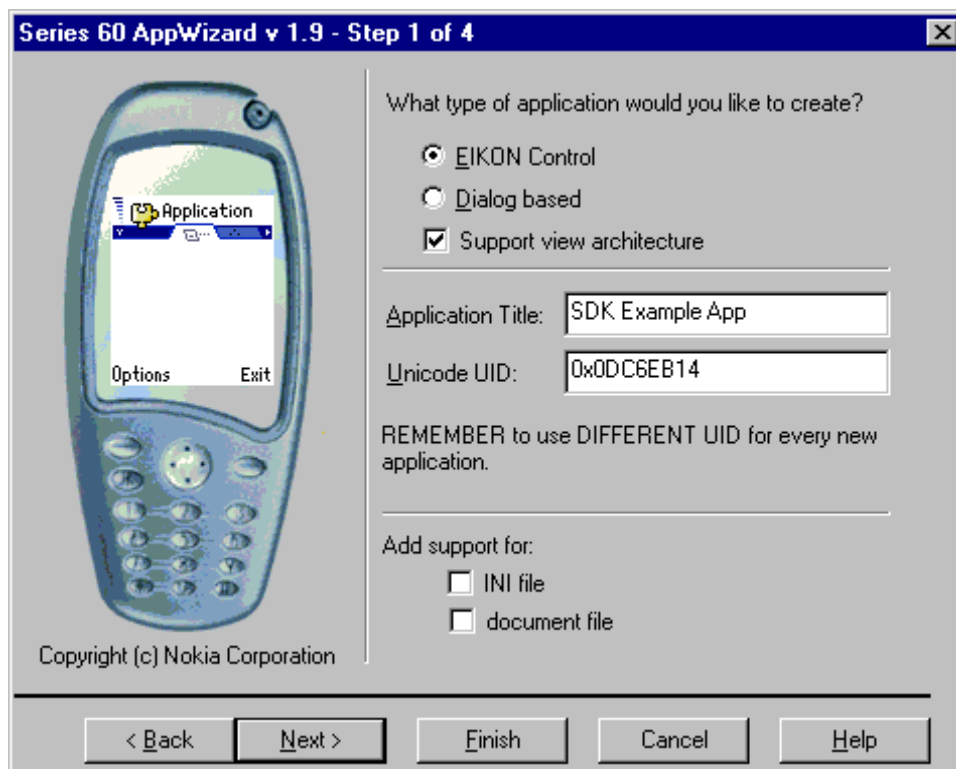
3.3.1 Vytvorenie novej aplikácie

Vytvorenie novej aplikácie začína založením nového projektu a definovaním pracovného adresára.

Je potrebné spustiť MS Visual C++ 6.0. Zvoliť File → New, a v následne zobrazenom okne zvoliť záložku Projects, tak sa v zobrazenom zozname zobrazí položka Series 60 AppWizard v 1.9 (viď obrázok 4.6). Keď na ňu klikneme myšou, tak je možné zvoliť názov projektu a pracovný adresár. Maximálna dĺžka názvu projektu je 16 znakov. Pracovný adresár je vhodné zvoliť na rovnakom disku ako je inštalácia SDK, vyhneme sa tak prípadným problémom. Po kliknutí na OK je toto dialógové okno ukončené.

3.3.1.1 Krok 1

V tomto kroku sa zobrazí prvé okno sprievodcu vytvorením aplikácie (z celkových štyroch). V tomto okne je možné definovať základné vlastnosti novej aplikácie (viď obrázok 3.9).



Obrázok 3.9: Prvé okno sprievodcu vytvorením novej aplikácie (prevzaté z [9]).

Najskôr sa zvolí architektúra aplikácie. Volíme medzi:

- **EIKON Control**
- **Dialog based**

Ak zvolíme **EIKON Control**, tak hlavný vzhľad aplikácie bude odvodený od triedy CCOECONTROL. Táto voľba je najčastejšia.

Ak zvolíme **Dialog based** tak vzhľad aplikácie bude odvodený od triedy CEIKDIALOG.

Ak zaškrtneme voľbu **Support view architecture**, sprievodca vytvorením aplikácie vytvorí dva náhľady. Každý obsahuje vzhľad podľa patričnej triedy na ktorej je založený.

Do poľa **Application Title** napíšeme názov, ktorý bude mať nová aplikácia. Skrátaná forma tohto názvu bude použitá aj pri generovaní názvov jednotlivých súborov projektu.

Každá aplikácia musí mať jedinečné UID. Sprievodca vytvorením novej aplikácie plní toto pole automaticky náhodnými hodnotami. Ale pre profesionálny vývoj aplikácií je možné požiadať firmu Symbian Ltd. o pridelenie jedinečného UID.

Nasledujúce dve voľby sa týkajú používaného súborového systému.

- **INI file** – vytvorí sa INI súbor pre aplikačné nastavenia
- **Document file** – vytvorí dokumentačný súbor pre aplikačné dáta.

Ak nie je potrebná podpora INI alebo Document súborov, tak tieto voľby necháme neaktívne. Tlačítkom **Next** prejdeme k ďalšej obrazovke.

3.3.1.2 Krok 2

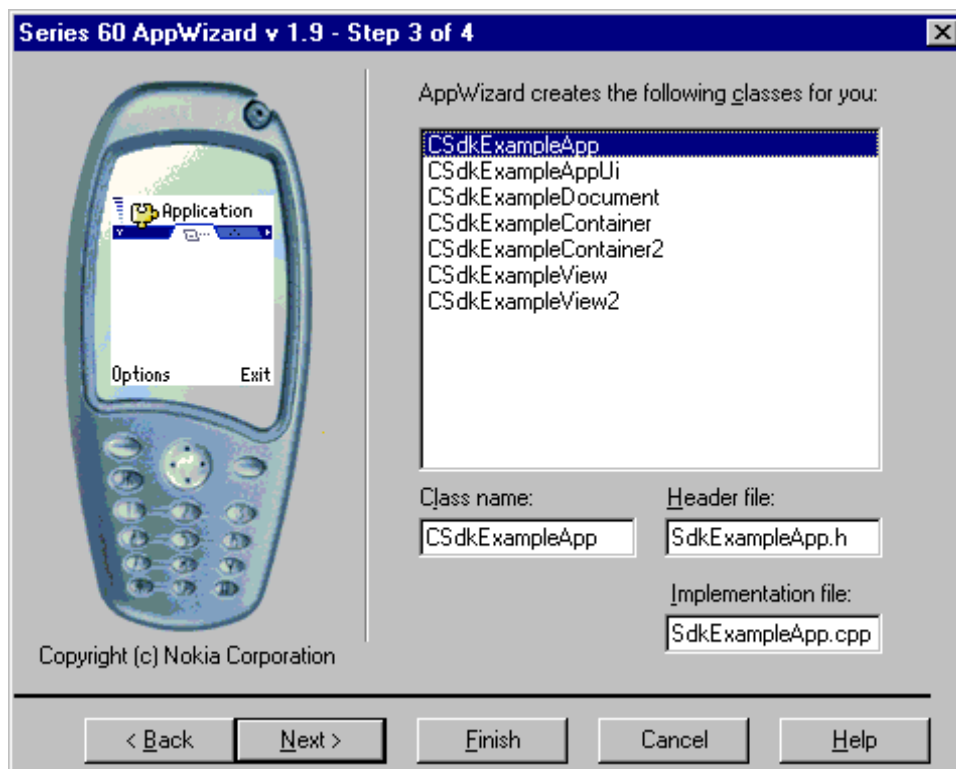
Na tejto obrazovke (vid'. obrázok 3.10) je možné zadať meno autora a informácie o copyrighte. Tieto informácie budú vložené na začiatok vygenerovaných zdrojových súborov.



Obrázok 3.10: Druhé okno sprievodcu vytvorením novej aplikácie (prevzaté z [9]).

3.3.1.3 Krok 3

Na tejto obrazovke (vid'. obrázok 3.11) je možné zmeniť implicitne zvolené mená pre triedy, hlavičkové a zdrojové súbory. Ak zvolíme niektorú triedu z uvedeného zoznamu, tak môžeme meniť jednotlivé názvy v editačných poličkach.



Obrázok 3.11: Tretie okno sprievodcu vytvorením novej aplikácie (prevzaté z [9]).

3.3.1.4 Krok 4

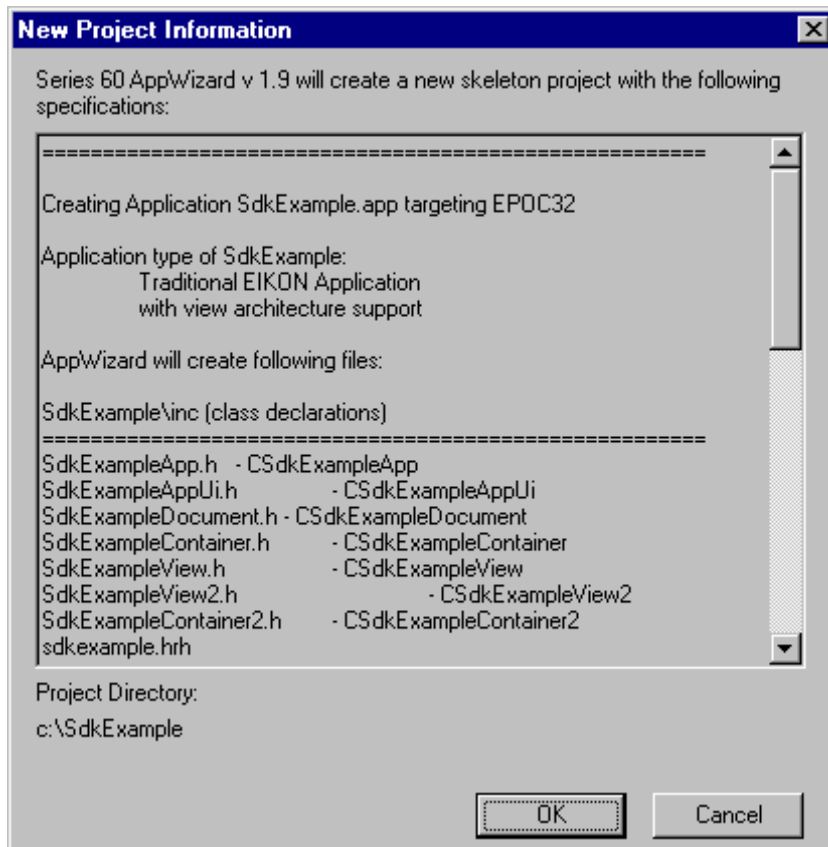
Na tejto obrazovke (vid'. obrázok 3.12) je možné zmeniť vygenerované názvy adresárov, ktoré tvoria logickú štruktúru projektu.



Obrázok 3.12: Štvrté okno sprievodcu vytvorením novej aplikácie (prevzaté z [9]).

3.3.1.5 Okno súhrnných informácií o projekte

Po zvolení voľby **Finish** na predošlej obrazovke sa zobrazí okno (viď. obrázok 3.13) so súhrnnými informáciami o novom projekte. Ak klikneme na **OK**, tak sa vytvorí nový projekt s uvedenými vlastnosťami. Nový projekt sa súčasne aj otvorí v MS Visual C++.



Obrázok 3.13: Okno so súhrnnými informáciami o novom projekte (prevzaté z [9]).

3.3.1.6 Zostavenie projektu

Tento nový projekt môže byť skompilovaný priamo vo MS Visual Studiu. Pre test využívajúci priložený emulátor vyberte v menu položku **Execute**. Visual Studio si vyžiada cestu k emulátoru (EPOC.EXE). Zvolíme napr. "c:\symbian\6.1\series60\epoc32\release\wins\udeb\epoc.exe". Týmto sa aplikácia skompiluje a spustí v prostredí emulátoru.

3.3.2 Aplikácia HelloWorld

Cieľom tejto kapitoly je ukážka tradičnej prvej aplikácie HelloWorld. Táto aplikácia po stlačení tlačítka menu vypíše na obrazovku telefónu „Hello“.

3.3.2.1 Aplikácia z pohľadu telefónu

Siemens SX1 je telefón, ktorý je osadený RISCovým procesorom ARM. Každá aplikácia, ktorú chceme na tomto telefóne používať musí byť skompilovaná pre túto platformu. Pri vývoji budeme ešte používať jednu virtuálnu platformu WINS, ktorá sa používa pre emulátor na počítači.

Typická Symbian aplikácia je tvorená týmito súbormi:

- helloworld.app – spustiteľný binárny súbor. Obsahuje strojový kód aplikácie.
- helloworld.rsc - resources binárny súbor. Obsahuje skompilované "resources" aplikácie – menu , dialógy, reťazce, atď.
- helloworld.aif - Application Information File. Obsahuje ikonu, názov aplikácie, atď.
- helloworld_caption.rss - resources binárny súbor. Obsahuje názov aplikácie (len u Series60)

Aplikačné súbory sa v telefóne Siemens SX1 nachádzajú v adresári C:\system\apps\<názov_aplikácie>. Prehliadač aplikácií si automaticky zaregistruje novú aplikáciu a pridá ju do svojej ponuky, ktorú vidíme na obrazovke telefónu. Ak sa nájde AIF súbor, priradí sa z neho ikona a názov aplikácie. Ak nie, tak systém priradí implicitnú systémovú ikonu a ako názov dosadí meno súboru aplikácie. Pri spustení aplikácie sa začne vykonávať kód aplikácie APP.

3.3.2.2 Aplikácia z pohľadu programátora

API Symbian OS a jeho rozšírenie Series60 umožňuje pri návrhu použiť tzv. Application Framework, ktorý zapúzdruje proces zavedenia aplikácie do pamäte a pridelenie systémových prostriedkov, vygenerovanie menu a vykreslenie hlavného okna aplikácie atď. Tento prístup je komplexný a vyžaduje dokonalé pochopenie a znalosti vnútorných väzieb medzi objektmi.

Pre začiatok si uvedieme štyri základné triedy aplikačného frameworku a ich najdôležitejšie väzby. Toto nám postačí k vytvoreniu helloworld aplikácie.

Sú to triedy: *CAknApplication*, *CAknDocument*, *CAknAppUi*, *CCoeControl*.

Životný cyklus typickej symbian aplikácie:

1. Operačný systém po spustení aplikácie volá funkciu *EXPORT_C CApaApplication* NewApplication()*, ktorá vytvára inštanciu triedy *CEikApplication* a vráti jej ukazateľ.
2. Pri inicializácii *CEikApplication* sa volá virtuálna funkcia *CreateDocumentL()* ktorá vytvára a vracia dokument aplikácie.
3. Dokument pri svojej inicializácii volá virtuálnu funkciu *CreateAppUiL()*, ktorá vytvára UI (user interface) *CEikAppUi* aplikácie.
4. UI obsahuje objekty pohľadov odvodených od triedy *CCoeControl*.
5. Trieda *CCoeControl* obsahuje virtuálnu funkciu *Draw(const TRect& aRect)*, ktorá vykreslí pohľad na obrazovku.
6. UI zachytáva vstupy od užívateľa a robí špecifické operácie nad objektmi aplikácie.
7. Užívateľ dáva prostredníctvom UI pokyn k ukončení aplikácie volaním funkcie . Dochádza k volaniu jednotlivých deštruktorov a k ukončeniu aplikácie.

3.3.2.3 Implementácia

Pomocou sprievodcu vytvorenia novej aplikácie si vytvoríme nový projekt. Vznikne nám napr. takáto adresárová štruktúra:

- C:\symbian\devel\helloworld\ – koreňový adresár
- C:\symbian\devel\helloworld\aiif – obsahuje definičný súbor AIF a ikony (.aiif, .bmp)
- C:\symbian\devel\helloworld\data – obsahuje resources a dátové položky aplikácie (.rss, ...)
- C:\symbian\devel\helloworld\group – obsahuje definičné súbory projektu (.mmp, .inf)
- C:\symbian\devel\helloworld\inc – obsahuje hlavičkové súbory (.h, .hrh)
- C:\symbian\devel\helloworld\sis – obsahuje definičný a inštalačný balíček SIS (.pkg, .sis)
- C:\symbian\devel\helloworld\src – zdrojové súbory (.cpp)

Trieda *CHelloWorldApplication* : *public CAknApplication*

Trieda *CHelloWorldApplication* dedí triedu *CAknApplication*, ktorá zapúzdruje hlavný aplikačný proces. Systém po spustení aplikácie volá funkciu *EXPORT_C CApaApplication* NewApplication()* (pozn.: obdoba funkcie *int main()*) v ktorej vytvárame inštanciu triedy *CHelloWorldApplication*. Implicitný konštruktor *CAknApplication()* uskutoční inicializáciu aplikácie low-level úrovni, prideluje systémové prostriedky a vytvára hlavný aplikačný proces.

```
EXPORT_C CApaApplication* NewApplication()
{
    return (new CHelloWorldApplication);
}
```

Po vytvorení aplikácie sa zavolá virtuálna funkcia *CApaDocument* CHelloWorldApplication::CreateDocumentL()*, aby získal ukazateľ na svoj dokument. V našom prípade funkcia vytvára dokument – inštanciu triedy *CHelloWorldDocument* volaním *CHelloWorldDocument::NewL()* a vráti naň ukazateľ. Každá aplikácia je tak v rámci frameworku priradený jej dokument.

```
CApaDocument* CHelloWorldApplication::CreateDocumentL()
{
    // Create an HelloWorld document, and return a pointer to it
    CApaDocument* document = CHelloWorldDocument::NewL(*this);
    return document;
}
```

Trieda *CHelloworldDocument*: public *CAknDocument*

Trieda predstavuje UI aplikácie, pomocou ktorého užívateľ komunikuje s aplikáciou. Pri vytváraní UI sa volá virtuálna funkcia *CAknAppUi::ConstructL()*, v ktorej by sme mali inicializovať UI zavolaním funkcie *CAknAppUi::BaseConstructL()*, ktorá zaisťuje načítanie aplikačných resources, generovanie menu, atp. Vo funkcií *CAknAppUi::ConstructL()* často inicializujeme aj iné položky. V našom prípade inicializujeme členskú premennú *iAppView*.

```
void CHelloWorldAppUi::ConstructL()
{
    BaseConstructL();
    iAppView = CHelloWorldAppView::NewL(ClientRect());
    AddToStackL(iAppView);
}
```

Virtuálna funkcia *CAknAppUi::HandleCommandL()* je volaná frameworkom ako reakcia na užívateľský vstup, typicky voľba položky v menu.

```
void CHelloWorldAppUi::HandleCommandL(TInt aCommand)
{
    switch(aCommand)
    {
        case EEikCmdExit:
        case EAknSoftkeyExit:
            Exit();
            break;
        case EHelloWorldCommand1:
            {
                _LIT(message, "Hello!");
                CAknInformationNote* informationNote = new (ELeave) CAknInformationNote;
                informationNote->ExecuteLD(message);
            }
            break;
        default:
            Panic(EHelloWorldBasicUi);
            break;
    }
}
```

Trieda *CHelloworldContainer*: *public CCoeControl*

Trieda predstavuje hlavný pohľad (view, container) aplikácie tj. okno, na ktoré môžeme kresliť, písať informácie, alebo umiestňovať iné GUI objekty typu *CCoeControl*. Trieda *CCoeControl* zapúzdruje základnú GUI komponentu, jej vlastnosti a chovanie. Jednou zo základných vlastností je možnosť kresliť na obrazovku v rámci svojho obdĺžniku, definovaným polohou a rozmermi komponenty. Vždy, keď systém alebo explicitne volaním funkcie *CCoeControl::DrawNow()* pošleme požiadavku na prekreslenie komponenty, zavolá sa virtuálna funkcia *CCoeControl::Draw()* const, ktorá prekreslí komponentu.

```
void CHelloWorldAppView::Draw(const TRect& /*aRect*/) const
{
    // Get the standard graphics context
    CWindowGc& gc = SystemGc();
    // Gets the control's extent
    TRect rect = Rect();
    // Clears the screen
    gc.Clear(rect);
}
```

Pohľad inicializujeme členskou funkciou *CHelloWorldAppView::ConstructL()*, v ktorej najskôr inicializujeme základné vlastnosti a chovanie komponenty volaním funkcie *CCoeControl::CreateWindowL()*, ďalej nastavíme obdĺžnik komponenty *CCoeControl::SetRect()* a nakoniec aktivujeme komponentu volaním funkcie *CCoeControl::ActivateL()*. Teraz sa nám pohľad zobrazí na obrazovke.

```
void CHelloWorldAppView::ConstructL(const TRect& aRect)
{
    // Create a window for this application view
    CreateWindowL();
    // Set the windows size
    SetRect(aRect);
    // Activate the window, which makes it ready to be drawn
    ActivateL();
}
```

Resources

Resources sa používajú k definovaniu menu, dialógov, reťazcov a iných statických štruktúr, ktoré aplikácia používa. Oddelenie týchto zdrojov od aplikácie je účelné napr. pri viacjazyčných verziách aplikácie, kde s inou jazykovou verziou dodávate len iné resources, aplikácia však zostáva rovnaká. Resources môžu byť taktiež zdieľané viacerými aplikáciami a tým šetria použité miesto. Resources sa definujú v .rss súboroch a po kompilácii sú uložené v binárnych súboroch .rsc.

Application Information File (.AIF)

AIF súbor popisuje aplikáciu v systéme, obsahuje informácie o aplikácii, ako je UID, názov, ikony a podporované MIME typy súborov. V skutočnosti sa jedná o zvláštny druh resources, ktorý definujeme v samostatnom .rss súbore pomocou jedinej štruktúry AIF_DATA. Pri kompilácii sa potom vytvorí meno_aplikácie.aif binárny súbor.

Project Definition File (.MMP)

Project Definition File .mmp je textový súbor, ktorý obsahuje popis projektu a jeho častí tak, aby bolo možné zostaviť Makefiles, pomocou ktorých aplikáciu a jej časti kompilujeme pre cieľovú platformu. Pri definícii projektu určujeme umiestnenie zdrojových a hlavičkových súborov aplikácie, knižnice ktoré aplikácia používa (.lib) a voči ktorým budeme linkovať, aif zdrojový súbor, resources atď.

Component description file (bld.inf)

Utilita bldmake používa súbor bld.inf k automatickému vygenerovaniu pomocného skriptu (dávkového súboru) abld.bat. Bld.inf uvádza v skupine PRJ_MMPFILES všetky projekty .mmp, ktoré chceme v priebehu "build-processu" používať (často sa jedná o jednu aplikáciu a viac DLL knižníc).

```
PRJ_MMPFILES
```

```
HelloWorld.mmp
```

3.3.2.4 Build-process a skript abld.bat

Skript abld.bat sa používa pre generovanie Makefiles a pre spúšťanie build-processu. Build-process spočíva v kompilácii, linkovaní a zostavení aplikácie a jej častí pre zvolenú platformu. Je vytvorený na základe bld.inf a v ňom uvedených .mmp súborov.

Build-process tvoria nasledujúce kroky:

1. Vytvorenie skriptu `abld.bat` a pravidiel pre vytváranie makefiles
2. Vytvorenie makefiles
3. Kompilácia pre cieľovú platformu

Vytvorenie `abld.bat`

Skript `abld.bat` vytvoríme z príkazového riadku nasledujúcim volaním:

```
c: \symbian\devel\helloworld\group>bldmake bldfiles
```

Mimo skriptu `abld.bat` v aktuálnom adresári sa nám vytvorili adresáre `%EPOCROOT%\epoc32\build\symbian devel\HELLOWORLD\group` so súbormi `.MAKE`, ktoré sa používajú ako pravidlá pre vytváranie Makefiles na zvolenú platformu

Vytvorenie Makefiles

Súbory Makefile uskutočňujú kompiláciu zdrojov, linkovanie a zostavenie aplikácie. Pri vytvorení Makefiles pre platformu ARM zavoláme `abld.bat` skript takto:

```
c: \symbian\devel\helloworld\group>abld.bat makefile arm
```

Po zavolaní sa v adresári

`%EPOCROOT%\epoc32\build\devel\symbian\HELLOWORLD\group` vytvoril podadresár `\armi` s makefile súborom `.ARMI`.

Ak chceme vytvoriť makefiles pre platformu WINS (emulátor) s podporou MS Visual C++ zavoláme:

```
c: \symbian\devel\helloworld\group>abld.bat makefile vc6
```

V adresári `%EPOCROOT%\epoc32\build\devel\symbian\HELLOWORLD\group` sa vytvorí adresár `\wins`, ktorý obsahuje makefile súbor `.MAKE` pre platformu WINS a projektové súbory `.DSP` a `.DSW` pre MS Visual C++.

Poznámka:

Takto môžeme vyvíjať aplikácie aj bez použitia MS Visual C++ a už čiastočne vytvorený projekt môžeme skompilovať tak, aby sa vytvorili aj projektové súbory so MS Visual Studio a tam potom pohodlne pokračovať v práci. Potom ale musíme mať na pamäti, že ak počas vývoja dôjde k

zmene projektového súboru .mmp (pridanie knižnice alebo zdrojového súboru), je nutné vždy vygenerovať nové Makefiles.

Kompilácia

Nakoniec skompilujeme a zostavíme aplikáciu pomocou vytvoreného skriptu `abld.bat` a `makefiles` následovne:

```
c: \symbian\devel\helloworld\group> abld.bat build armi urel
```

Skompilované aplikačné súbory `helloworld.app`, `helloworld.rsc`, `helloworld_caption.rsc`, `helloworld.aif` nájdeme v adresári `%EPOCROOT%\epoc32\release\armi\urel`.

Objektové súbory `.o` nájdeme v adresári:

```
%EPOCROOT%\epoc32\build\symbian\devel\HELLOWORLD\group\urel.
```

Vytvorenie inštalačného balíčku SIS

Inštalačný balíček SIS je binárny súbor, ktorý sa používa pre zapúzdrenie, distribúciu a korektnú inštaláciu aplikačných súboru na telefón. Ďalej obsahuje informácie o aplikácii a umožňuje vkladanie certifikátov pre overenie pravosti aplikácie. SIS balíček je možné nainštalovať z PC alebo priamo z telefónu. Balíček definujeme v súbore `.pkg`.

Pre vytvorenie SIS balíčka použijeme utilitu `makesis` takto:

```
cd c:\symbian\devel\helloworld\sis
```

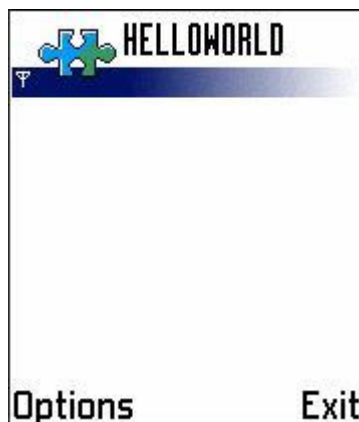
```
c:\symbian\devel\helloworld\sis >makesis helloworld.pkg
```

Poznámka:

Celá aplikácia `helloworld` sa nachádza na priloženom CD. V prípade nejasností a pre vytvorenie uceleného pohľadu o tejto aplikácii je vhodné si túto aplikáciu preštudovať a skúsiť si na nej celý build process.

3.3.2.5 Screenshoty aplikácie HelloWorld

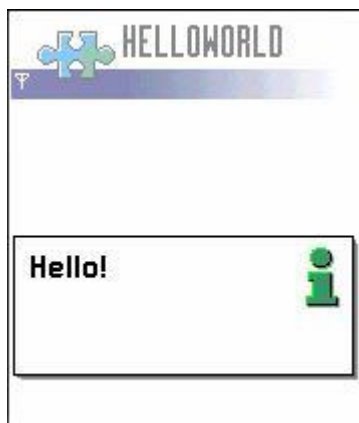
Tu sú uvedené jednotlivé obrazovky aplikácie HelloWorld. (viď obrázky 3.14, 3.15, 3.16)



Obrázok 3.14: Okno aplikácie HelloWorld po štarte (prevzaté z [9]).



Obrázok 3.15: Okno aplikácie HelloWorld – zobrazenie menu (prevzaté z [9]).



Obrázok 3.16: Okno aplikácie HelloWorld – zobrazenie Hello!(prevzaté z [9]).

4 Analýza súčasného stavu

Obsahom tejto kapitoly je preskúmanie niektorých existujúcich aplikácií pre mobilné telefóny s podporou OS Symbian, ktoré používajú integrovaný fotoaparát v mobilnom telefóne na zachytenie obrazu a jeho následné spracovanie. Najčastejšími aplikáciami tohto druhu sú rôzne editory a prehliadače obrázkov a programy typu „fotoaparát“.

4.1 Popis a podporované funkcie

Väčšina aplikácií pre mobilné telefóny s OS Symbian s podobným zameraním sú po vzhľadovej stránke veľmi podobné. Je to dané tým, že väčšina programátorov pri vývoji používa štandardné ovládacie prvky. Medzi ne patrí napríklad: menu, dialógy, zoznamy, editačné prvky a záložky. Táto skutočnosť prispieva k tomu, že sa užívateľ dokáže veľmi rýchlo zorientovať a používať aj nový program intuitívne a bez vážnejších problémov.

4.1.1 Aplikácie umožňujúce fotografovanie

Hlavné funkcie (často jediné) sú zachytenie snímky z fotoaparátu a jej následné zobrazenie na display a uloženie do súboru. Medzi typických predstaviteľov tejto skupiny patrí aplikácia s názvom „Momentka“ (viď. obrázok 4.1), ktorá je štandardne súčasťou programového vybavenia mobilného telefónu Siemens SX1.



Obrázok 4.1: Vzhľad aplikácie Momentka z mobilného telefónu Siemens SX1.

4.1.2 Prehliadače obrázkov

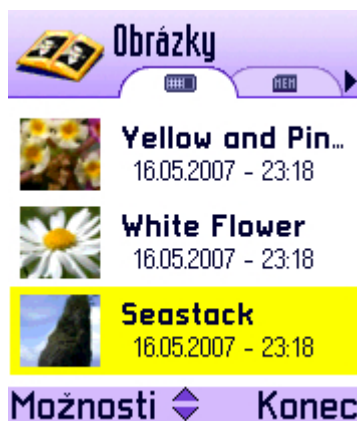
Ako z názvu vyplýva, je hlavnou funkciou zobrazovať obrázky, ktoré sú uložené v pamäti mobilného telefónu. Existuje niekoľko koncepcií, z ktorých sa najčastejšie používajú dve.

Keďže je potrebné nejakým spôsobom určiť, kde je obrázok fyzicky uložený, tak prvou možnosťou je vždy po spustení aplikácie vyhľadať všetky obrázky. Následne sa zobrazia náhľady na tieto obrázky a prostredníctvom týchto náhľadov je potom možné jednotlivé obrázky zobrazit'. Tento prístup je často až nepríjemne pomalý, preto existuje modifikácia, ktorá obrázky vyhľadáva len v určitých adresároch (napr. E:\Images). Takto sa dosiahne zrýchlenia, ale na druhú stranu obmedzíme univerzálnosť takejto aplikácie. Príklad takýchto aplikácií je na nasledujúcom obrázku.



Obrázok 4.2: Ukážka prehliadačov obrázkov prvého typu.

Druhý typ (viď. obrázok 4.3) je založený na koncepcii súborového manažéra. Zobrazuje vždy obsah jedného adresára a pre každý súbor obsahujúci obrázok zobrazí jeho náhľad. Tento koncept môže byť principiálne o trochu výkonnejší ako predošlý. A skutočnosť, že užívateľ presne vie, kde je daný obrázok uložený, môže byť veľmi užitočná. Vzhľadovo sa môže podobat' kombinácií nasledujúcich dvoch obrázkov.



Obrázok 4.3: Možná podoba prehliadača obrázkov druhého typu.

4.1.3 Editory obrázkov

Editormi obrázkov rozumieme aplikácie, ktoré dokážu obrázok určitým spôsobom meniť. Na obrázku je možný vzhľad jednoduchého editoru a niekoľko základných funkcií, ktoré môže poskytovať.



Obrázok 4.4: Jednoduchý editor obrázkov s funkciou rotácie a zväčšenia obrázku

4.2 Zhrnutie kapitoly

Uvedené príklady aplikácií čiastočne reprezentujú možnosti práce s integrovaným fotoaparátom a s obrázkami, ktoré sú už uložené v pamäti mobilného telefónu. Tieto aplikácie majú často úzko špecifikovanú funkčnosť (napr. jedna len zobrazuje obrázky, iná dokáže odfotografovať obrázok a uložiť ho na disk).

Bežný užívateľ by často radšej uvítal viacej funkcií v jednej aplikácii ako viac aplikácií s obmedzenou funkčnosťou. Preto v nasledujúcej kapitole navrhujeme komplexnejšiu aplikáciu, ktorá bude zahrňovať funkcie aplikácií spomenutých v tejto kapitole.

5 Návrh aplikácie – inteligentný editor obrázkov

Ako vhodná aplikácia pre reprezentáciu možností spracovania obrazu na mobilných telefónoch s OS Symbian bola zvolená aplikácia „Editor obrázkov“. Pretože aplikácie spomenuté v predošlej kapitole boli v niektorých prípadoch výkonnostne dosť obmedzené, budeme sa snažiť navrhnúť výkonnostne robustnejšiu aplikáciu. Zameriame sa hlavne na vyriešenie výkonnostných problémov funkcií zoom a náhľad obrázkov.

5.1 Popis navrhovaných funkcií

Aplikácia editor obrázkov bude navrhovaná a testovaná pre smartphone Siemens SX1. Z toho vyplýva, že operačným systémom bude Symbian OS vo verzii 6.1.

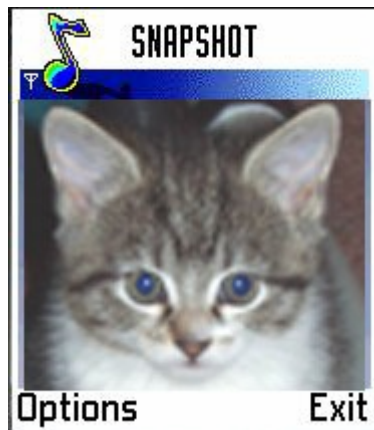
Základné požiadavky kladené na túto aplikáciu:

- zachytenie snímku z integrovaného fotoaparátu
- uloženie odfotografovaného snímku do súboru
- načítanie obrázku zo súboru
- rotácia obrázku o 90, 180 a 270 stupňov
- funkcia zoom – zväčšenie obrázku
- zobrazovanie náhľadu na obrázky v jednotlivých adresároch
- prevod farebného obrázku na čiernobiely (stupne šedi)
- prípadne ďalšie obrazové operácie

5.1.1 Zachytenie snímku z integrovaného fotoaparátu

Pre túto časť aplikácie budú využité funkcie, ktoré obsahujú triedy CCameraManager a MCameraObserver, ktoré sú súčasťou dodávaných knižníc a slúžia na prácu s integrovaným fotoaparátom.

Snahou bude doceliť aspoň čiastočne komfortu obsluhy integrovaného fotoaparátu známeho z podobných aplikácií pre mobilné telefóny. Výsledok by sa mohol podobáť nasledujúcemu obrázku.



Obrázok 5.1: Okno zobrazujúce obrázok z integrovaného fotoaparátu (prevzaté z [9]).

5.1.2 Zobrazenie náhľadu v jednotlivých adresároch

Dostupné aplikácie ponúkajú možnosť zobrazovať súbory s obrázkami v jednotlivých adresároch takým spôsobom, že zobrazujú v podobe obrazovej ikony náhľad na obrázok uložený v príslušnom súbore. Nevýhodou takéhoto zobrazenia je fakt, že takéto zobrazovanie trvá dosť značnú dobu a znehodnocuje inak veľmi príjemnú podobu zobrazenia jednotlivých súborov. Hlavnú časť doby, ktorú trvá zobrazenie náhľadu zaberá otvorenie súboru s obrázkom a vytvorenie obrazovej ikony.

Navrhovaným riešením je preto funkcia, ktorá by umožňovala hromadné uloženie zmenšených náhľadov priamo do rovnakého adresára, kde sú uložené jednotlivé obrázky v originálnom rozlíšení. Výsledkom bude adresár, ktorý bude obsahovať pôvodné súbory s obrázkami a navyše v tomto adresári pribudne adresár s presne špecifikovaným názvom. Do tohto adresára sa ku každému obrázku vytvorí súbor s rovnakým menom ako je názov súboru s príslušným obrázkom. V tomto súbore bude uložený zmenšený obrázok s rozlíšením niekoľko málo obrazových bodov. Následne už bude aplikácia pri zobrazovaní náhľadov zobrazovať pripravené zmenšené obrázky čo bude menej časovo náročné. Takto v podstate dosiahneme funkčnosti, ktorú poskytujú aplikácie typu súborový manažér, čo je veľmi praktické.

Tieto zmenšené obrázky bude možné vytvárať automaticky napríklad pre všetky obrázky v jednom adresári alebo prípadne aj v zanorených adresároch. Nabáda sa však otázka, čo sa stane, ak niekde pribudne nový obrázok a nebude k nemu vytvorená jeho zmenšenina? Existujú v podstate dve riešenia.

Prvým je snaha aplikácie takýto obrázok počas zobrazovania jeho náhľadu najskôr takýto obrázok otvoriť a vytvoriť si jeho náhľad, ktorý sa potom zobrazí. Druhým riešením je pri nájdení obrázku, ktorý nemá vytvorený svoj náhľad, zobrazit' nejakú jednotnú ikonu, ktorá reprezentuje ľubovoľný obrázok. A náhľad k takémuto obrázku by sa vytvoril až pri jeho fyzickom otvorení, alebo po vyvolaní funkcie hromadného vytvorenia náhľadov.

Grafická vzhľad tejto časti aplikácie bude pravdepodobne podobný vzhľadu na obrázku 5.3.

5.1.3 Načítanie a uloženie obrázku z a do súboru

Obrázok pre spracovanie môžeme okrem odfotografovania získať aj otvorením zo súboru. Po prevedených úpravách obrázku bude samozrejme možné uložiť výsledný obrázok do súboru.

Predpokladané podporované obrazové formáty sú BMP, JPEG, atď.

Predpokladaný spôsob otvorenia obrázku bude realizovaný pomocou vytvoreného súborového manažéra popísaného v kapitole 6.1.2. Užívateľ si s jeho pomocou vyberie požadovaný súbor a po jeho zvoľení pomocou niektorej klávesy na mobile sa tento súbor otvorí a obrázok sa zobrazí na displeji. Uloženie obrázku bude realizované jednoduchším spôsobom. Všetky obrázky sa budú ukladať vždy na rovnaké miesto. Presne to bude adresár s názvom Images na pamäťovej karte, ktorá je súčasťou mobilného telefónu. Takto sa správajú aj niektoré iné dostupné aplikácie.

5.1.4 Rotácia a zväčšenie obrázku

Ďalšou funkciou bude rotácia obrázku. Momentálne sa predpokladá rotácia o 90, 180 a 270 stupňov.

Funkcia zoom bude daný obrázok schopná zväčšiť. Všetky tieto funkcie sú podporované triedou MdaImageUtilObserver, ktorá je súčasťou dodávaných knižníc.

Avšak problémom niektorých existujúcich aplikácií s funkciou zväčšenie obrázku je skutočnosť, že pri práci s obrázkom s väčším rozlíšením (cca viac ako 2 Mpx) pri viacnásobnom zväčšení aplikácia dospeje až do bodu, kedy mobilný telefón zobrazí informáciu o nedostatku operačnej pamäte. Táto skutočnosť je dosť neprijemná, preto by bolo vhodné funkciu zoom implementovať vhodnejším spôsobom, ktorý je menej náročný na využitie operačnej pamäte.

Momentálna idea je zväčšovať len tú časť obrázku, ktorá je práve zobrazovaná na displeji mobilného telefónu. Túto oblasť ale pre zvýšenie efektívnosti rozšírime na každú stranu o určitý počet pixelov (napr. o cca štvrtinu rozlíšenia displeja telefónu). Toto je následne výhodné pri posuve zobrazovaného obrázku, pretože pri jemnom posune stačí bitmapu skutočne len posunúť a nemusí okamžite dôjsť k prepočítavaniu celej scény. K tomu by došlo až pri výraznejšom posune.

5.1.5 Prevod farebných obrázkov na čiernobiele (stupne šedi)

Táto známa funkcia prevedie redukciu farieb obrazu na odtiene šedi. Farebné odtiene môžu byť na odtiene šedi prevedené pomocou vzorca:

$$I = 0,299 * R + 0,587 * G + 0,114 * B$$

kde: I – je výsledná intenzita (úroveň šedi)

R, G, B – sú základné farebné zložky pôvodnej farby

Prevod obrazu na úrovně šedi spočívá v postupném prepočítaní jednotlivých bodov obrazu podľa tohto vzorca.

5.1.6 Iné obrazové operácie

Do aplikácie budú podľa uváženia pridané ešte iné obrazové operácie.

5.2 Zhrnutie kapitoly

V tejto kapitole vznikol návrh aplikácie, ktorá bude pracovať s obrázkami. Obrázok získa buď z integrovaného fotoaparátu alebo otvorením zo súboru. Na prehliadanie jednotlivých obrázkov posluží súborový manažér, ktorý dokáže zobrazovať aj náhľady na jednotlivé obrázky. Vybraný súbor s obrázkom potom dokáže otvoriť a zobraziť. Na takto pripravený obrázok môžeme aplikovať jednotlivé funkcie. Napríklad rotáciu alebo ho previesť na stupne šedej a takto upravený obrázok následne uložiť do súboru.

6 Vlastná realizácia

Táto kapitola pojednáva o realizácii navrhutej aplikácie. V prvej časti popisuje celú jej štruktúru a predpoklady, ktoré musia byť splnené pre správnu funkciu vývojového prostredia a následne celej aplikácie. V druhej časti sa pozrieme na dostupné funkcie a vzhľad aplikácie.

6.1 Štruktúra aplikácie a vývojové nástroje.

Pre úspešný vývoj akejkoľvek aplikácie je správne nastavené vývojové prostredie a všetky používané nástroje. Aby sa zamedzilo prípadným nedorozumeniam, uvediem nastavenie, ktoré som úspešne používal ja.

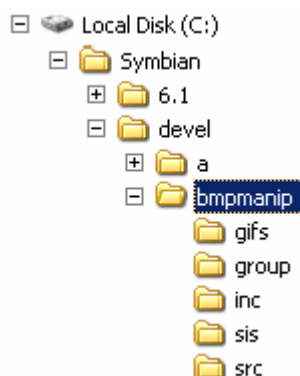
6.1.1 Nastavenie vývojových nástrojov

Prvým predpokladom je správne nainštalované SDK a to tak, ako je popisované v kapitole 3.2.2 a 3.2.3. Pre správnu funkciu je mimo inštalácie nutné aj správne nastavenie systémových premenných. Funkčnosť je vhodné overiť na nejakom jednoduchom príklade. Vhodným kandidátom je aplikácia HelloWorld, ktorá sa nachádza na priloženom CD a postup celého procesu kompilácie a zostavenia tejto aplikácie je v kapitole 3.3.2

6.1.2 Štruktúra realizovanej aplikácie

Štruktúra celej aplikácie sa prísne podriadiť presným požiadavkám, ktoré musí spĺňať aplikácia pre Symbian OS. Obsah niektorých súborov popisovaných ďalej je možné nájsť v prvej prílohe. Všetky súbory sú potom dostupné na priloženom CD.

Pretože pri kompilácii celej aplikácie sa využívajú cesty, ktoré sú nastavené v súbore *bmpmanip.pkg*, doporučujem adresár s názvom *bmpmanip*, ktorý obsahuje celú výslednú aplikáciu uložiť do adresára: *c:\Symbian\devel*. Názorne to popisuje následný obrázok.



Obrázok 6.1: Umiestnenie a adresárová štruktúra aplikácie Bmpmanip.

Výsledná adresárová štruktúra a popis jednotlivých podadresárov:

C:\symbian\devel\bmpmanip\ – koreňový adresár

C:\symbian\devel\bmpmanip\gifs – obsahuje obrázky používané aplikáciou

C:\symbian\devel\bmpmanip\group – obsahuje definičné súbory projektu (.mmp, .inf, .rss)

C:\symbian\devel\bmpmanip\inc – obsahuje hlavičkové súbory (.h, .hrh)

C:\symbian\devel\bmpmanip\sis – obsahuje definičný a inštalačný balíček SIS (.pkg, .sis)

C:\symbian\devel\bmpmanip\src – zdrojové súbory (.cpp)

V adresári *group* je umiestnený aj súbor *go.bat*. Tento dávkový súbor obsahuje (obrázok 6.2) všetky potrebné príkazy na úplnú kompiláciu a zostavenie výslednej aplikácie. Je ale nutné dodržať nastavenie vývojových nástrojov a umiestnenie projektu presne tak, ako je popísané v tejto kapitole.

```
cd c:\symbian\devel\bmpmanip\group
call bldmake bldfiles
cd c:\symbian\devel\bmpmanip\group
call abld.bat makefile armi
cd c:\symbian\devel\bmpmanip\group
call abld.bat build armi urel
cd c:\symbian\devel\bmpmanip\sis
call makesis bmpmanip.pkg
cd c:\symbian\devel\bmpmanip\group
```

Obrázok 6.2: Príkazy obsiahnuté v dávkovom súbore *go.bat*

Podrobnejší pohľad na tento súbor nám prezradí postupnosť potrebného procesu na zostavenie výsledného inštalačného balíčka s názvom *bmpmanip.sis*, ktorý sa v prípade úspechu vytvorí v adresári *C:\symbian\devel\bmpmanip\sis*.

Na začiatku sa volá *bldmake bldfiles*, tento príkaz po ukončení vytvorí v adresári *C:\symbian\devel\bmpmanip\group* dávkový súbor *abld.bat*, ktorý sa potom volá v ďalších riadkoch. Význam nasledujúcich príkazov, tak ako idú po sebe, je vytvorenie tzv. „makefiles“ pre architektúru ARMI, následne z tohto zostavenie aplikácie. A úplne nakoniec vytvorenie výsledného inštalačného balíčka podľa definície v súbore *bmpmanip.pkg*.

Ak chceme docieľiť vytvorenie makefiles pre platformu WINS (emulátor) s podporou MS Visual C++ zavoláme:

```
c:\symbian\devel\bmpmanip\group>abld.bat makefile vc6
```

V adresári

C:\Symbian\6.1\Series60\Epoc32\BUILD\SYMBIAN\DEVEL\BMPMANIP\GROUP\BMPMANIP sa vytvorí adresár *\wins*, ktorý obsahuje makefile súbor *.MAKE* pre platformu WINS a projektové súbory *.DSP* a *.DSW* pre MS Visual C++. Od tejto chvíle môžeme otvoriť celý projekt s využitím MS

Visual C++ a využívať všetky dostupné výhody tohto vývojového prostredia. Jedným z najdôležitejších výhod je priama možnosť kompilácie a spustenia aplikácie na emulátore. Popis práce s emulátorom je v kapitole 3.3.1.6. Na emulátore nefunguje funkcia fotografovania obrázku. Pre vyskúšanie tejto funkcie je nutné nainštalovať aplikáciu do mobilného telefónu. K tomuto slúži práve súbor *bmpmanip.sis*. Tento súbor presunieme do mobilného telefónu a tam ho dáme nainštalovať. Alebo pre tento účel použijeme aplikáciu, ktorú štandardne dodáva výrobca mobilného telefónu. Nazýva sa väčšinou ako „manažér telefónu“ a umožňuje správu a obsluhu telefónu prostredníctvom osobného počítača.

V prípade, že je projekt umiestnený inde, ako v *C:\symbian\devel\bmpmanip* nejde samozrejme použiť dávkový súbor *go.bat*. V tomto prípade je nutné postupovať po jednotlivých príkazoch spúšťaných z patričných adresárov (podľa skutočného umiestnenia projektu). A je možné, že bude potrebné upraviť aj súbor *bmpmanip.pkg*.

6.2 Vzhľad a popis aplikácie bmpmanip

Postupne sa dostávame ku vzhľadu a popisu jednotlivých funkcií aplikácie. Pri každej popísanej funkcii bude napísaný názov funkcie v súbore *bmpmanipappview.cpp*, ktorá sa vyvolaním príslušnej funkcie vykoná. Obsah tohto súboru je uvedený v prvej prílohe. Pre komplexné preskúmanie zdrojových kódov tejto aplikácie je vhodnejšie prezrieť všetky súbory dostupné na priloženom CD.

6.2.1 Dostupné funkcie a ovládanie aplikácie

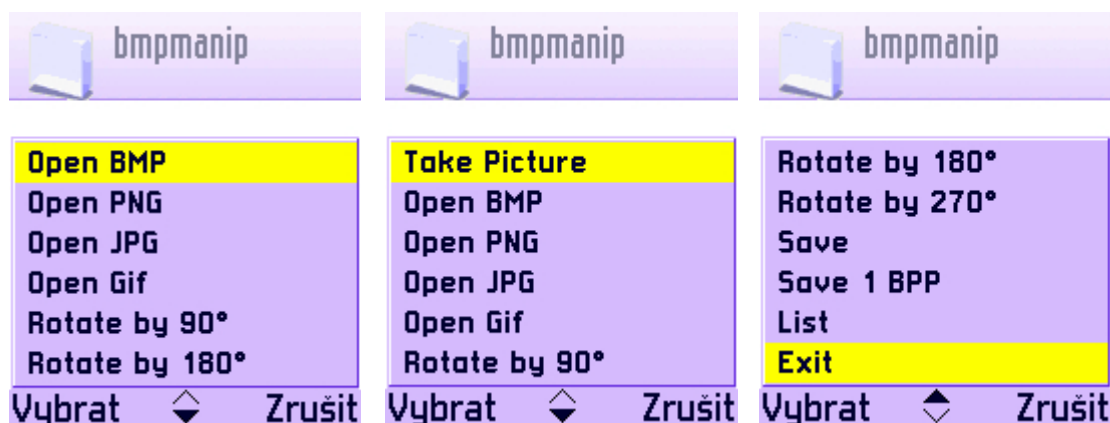
Po spustení aplikácie sa zobrazí okno aplikácie klasické pre Symbian OS. Jeho vzhľad je na obrázku 6.3. V hornej časti okna je zobrazená ikona a názov aplikácie, pod ním sa nachádza priestor, ktorý permanentne slúži na vykresľovanie všetkých informácií a ovládacích prvkov. Úplne dole je úzka lišta s dvojicou textových polí. Text popisuje význam funkčných kláves umiestnených tesne pod displejom mobilného telefónu. Toto je štandardný prístup vo všetkých aplikáciách.



Možnosti Konec

Obrázok 6.3: Vzhľad aplikácie tesne po spustení. Dole je vidieť význam funkčných kláves.

Po zvolení možnosti koniec sa aplikácia štandardne ukončí. Voľba „Možnosti“ vyvolá roletové menu s dostupnými funkciami (obrázok 6.4). V tomto menu sa dá pohybovať pomocou joysticku. Po potvrdení vybranej funkcie sa táto vykoná, ak je to možné.



Obrázok 6.4: Kompletné menu aplikácie.

Na obrázku úplne vľavo je vidieť menu, keď ešte nie je inicializovaný integrovaný fotoaparát. Vtedy nie je možné fotografovať, preto je príslušná položka menu skrytá. Akonáhle sa kamera inicializuje, je v menu zobrazená položka „Take Picture“ a je možné fotografovať. Na obrázku úplne vpravo je druhá časť menu.

6.2.2 Funkcia fotografovania a otvárania obrázkov

Ak už je v menu zobrazená funkcia „Take Picture“, je možné získavať obrázky z integrovaného fotoaparátu. Po zvolení sa odfotografuje scenéria, na ktorú je práve zameraný objektív. V zdrojovom kóde tomu odpovedá funkcia *Snap()* (je možné nájsť v prílohe číslo 1). Po odfotografovaní sa pomocou funkcie *Draw()* vykreslí získaný obrázok na display telefónu.

Všetky položky menu začínajúce slovom „Open“ majú za účel otvoriť obrázok uložený v súbore v pamäti mobilného telefónu. Skratka za slovom Open upresňuje, aký typ súboru sa bude otvárať. Aplikácia dokáže otvoriť súbory s príponou .BMP, .JPG, .PNG, .GIF. Súbory s obrázkami v príslušnom formáte sú uložené v adresári, kde je nainštalovaná aplikácia. Konkrétne je to adresár *C:\System\Apps\bmpmani*. V zdrojovom kóde tomuto zodpovedá funkcia *OpenPicture()*. Po úspešnom otvorení sa uskutoční konverzia obrázku do bitmapy. Obrázku sa ďalej zmení rozmer. Zistí sa či je obrázok širší alebo vyšší. Podľa toho sa rozmer bitmapy upraví tak, že dlhšia strana sa zmení na 150 pixelov a kratšia sa prispôbi podľa pôvodného pomeru strán. V zdrojovom kóde je to funkcia *ScaleL* s príslušnými parametrami. V konečnom dôsledku to znamená, že obrázky *image1.bmp* a *image1.jpg* sa musia zväčšiť a obrázky *image1.png* a *image1.gif* sa musia zmenšiť. Takto upravený obrázok sa následne zobrazí. Na obrázkoch 6.5 a 6.6 je možné vidieť varianty, ktoré môžu pri otváraní obrázku nastať.



Obrázok 6.5: Otvorenie obrázku image1.bmp a image1.jpg



Obrázok 6.6: Otvorenie obrázku image1.png a image1.gif. Chyba pri otváraní obrázku.

6.2.3 Rotácia obrázkov

Pomocou položiek menu začínajúcich slovom „Rotate“ je možné otočiť obrázok, ktorý je práve zobrazený, o 90°, 180° a 270°.

V programovom kóde tomu odpovedá funkcia *RotateImageClockwiseL* volaná s príslušnými parametrami. Na nasledujúcich obrázkoch je vidieť rotáciu jedného z obrázkov.



Obrázok 6.7: Rotácia obrázku o 90°, 180° a 270° oproti originálu.

6.2.4 Uloženie obrázkov

Nakoniec je možné práve zobrazovaný obrázok uložiť do súboru. Obrázok sa uloží v takej pozícii ako ho práve vidíte. Ale rozlíšenie obrázku bude zodpovedať veľkosti obrázku po konverzii a nie veľkosti ktorú vidíme na mobilnom telefóne.

Ukladanie obrázkov zabezpečujú položky menu „Save“ a „Save 1 BPP“. Položke „Save“ odpovedá funkcia *SaveL()* a pri voľbe „Save 1 BPP“ sa v skutočnosti volá funkcia *Save1BppL()*. Obrázok sa v oboch prípadoch uloží do súboru *C:\System\Apps\bmpmani\newimage.bmp* v mobilnom telefóne. V prvom prípade sa obrázok uloží ako bitmapa s 24 bitovou farebnou hĺbkou. V druhom prípade som len pre zaujímavosť zvolil farebnú hĺbkou 1 bit na pixel.

6.2.5 Ostatné položky menu

V menu nám zostali ešte dve nepopísané funkcie. Po zvolení možnosti „List“ bolo plánované, že sa zobrazí plánovaný súborový manažér s funkciou náhľadov na obrázky. V skutočnosti sa volá funkcia *ListL()*. Momentálne ale obsahuje iba kód pre zobrazenie správy o tom, že funkcia nieje implementovaná. Presná príčina bude popísaná v kapitole 7.

Úplne poslednou položkou menu je „Exit“. Po jej zvolení sa aplikácia korektne ukončí.

6.3 Zhrnutie kapitoly

Začiatok tejto kapitoly obsahuje popis štruktúry implementovanej aplikácie a určité predpoklady, ktoré je potreba dodržať, aby kompilácia a následné zostavenie aplikácie prebehlo bez problémov.

Na túto časť naväzuje popis ovládania aplikácie a pohyb v jeho menu. Následne pojednáva o jednotlivých funkciách a vzhľade aplikácie v priebehu vykonávania jednotlivých úloh. Medzi funkcie patrí možnosť získania obrázku z fotoaparátu alebo zo súboru. Na to naväzujú funkcie

7 Výsledky

Počas realizácie tohto projektu postupne dochádzalo ku sledu udalostí, ktoré predstavovali určité problémy a priamo ovplyvnili dosiahnuté výsledky.

7.1 Analýza problémov

Podme sa bližšie pozrieť na jednotlivé problémy spojené s týmto projektom. Ich postupné riešenie do značnej miery ovplyvnili konečnú podobu tejto práce.

Na začiatku som zápasil hlavne s nedostatkom informácií a literatúry o tejto problematike. OS Symbian je relatívne nový trend, takže tomu odpovedalo aj množstvo dostupných informácií. Po pochopení princípu vývoja aplikácií pre tento operačný systém a získania správnej verzie SDK pre použitý mobilný telefón bolo možné začať s pokusmi vytvoriť vlastnú aplikáciu.

V zadaní tejto práce sa kladie dôraz na preskúmanie možností vývoja aplikácií pre OS Symbian pre spracovanie obrazu a videa z integrovanej kamery. Preto moje prvé snaženie smerovalo k tejto méte. Bolo potrebné odfotografovať obrázok a v mobilnom telefóne ho spracovať a následne uložiť. Získať obrázok z fotoaparátu nebol až taký problém, ale problém nastal, keď bola potreba uložiť obrázok do súboru. Uložiť ho inak ako klasickú bitmapu (napr. JPEG, PNG, atď.), sa ukázalo ako vážny problém. Rôznymi pokusmi na toto téma som strávil desiatky, možno stovky hodín. Problém je, že to neprinieslo skoro žiadny výsledok. Uspokojivý spôsob ako prekonvertovať bitmapu do JPEGu som našiel až v API pre Symbian verzie 7. Ja som použil mobilný telefón s OS Symbian verzie 6.1, takže tento spôsob nebol použiteľný.

Preto som sa radšej začal venovať načítaniu obrázkov zo súboru. Postupne vznikla aplikácia, ktorá dokázala fotografovať a otvárať obrázky zo súborov. A následne ich uložiť ako bitmapu do súboru. Postupne sa pridala schopnosť vykresliť ich na display mobilného telefónu a zmeny rozlíšenia obrázku. Nasledovali funkcie rotácie o 90°, 180 ° a 270°.

Ďalšou plánovanou etapou bol súborový manažér s možnosťou vytvárania náhľadov na obrázky. V kapitole 5.1.2 je presne popísaný plánovaný princíp. Ako vhodným prostriedkom k dosiahnutiu žiadaného výsledku boli zvolené zoznamy (List Boxes). Je ich viacej druhov, pre svoje vlastnosti bol vybraný zoznam s položkami, ktoré dokážu zobrazit' obrazovú ikonu a dvojriadkový text (vid' obrázok 7.1). V nájdených príkladoch je aj uvedený spôsob použitia. Má ale pár nedostatkov. Tento zoznam sa vytvára v podstate staticky pomocou nadefinovania jeho vzhľadu v súbore zdrojov (resource súbory). Vytváranie jeho vzhľadu dynamicky za behu programu sa mi tiež nepodarilo uspokojivo vyriešiť.



Obrázok 7.1: Zvolený typ zoznamu, ktorý vyhovuje požiadavkám na tvorenie náhľadov na obrázky (zelené G je obrazová ikona).

7.2 Vyhodnotenie výsledkov

V dôsledku vzniknutých problémov, popísaných v predošlej kapitole, nemá výsledná aplikácia všetky pôvodne plánované funkcie. Ale napriek tomu aplikácia ukazuje spôsoby spracovania obrazu na mobilných telefónoch s OS Symbian. Ukazuje základné zásady, ktoré treba dodržiavať pri vývoji aplikácií. Je potrebné dodržiavať základnú štruktúru aplikácie. Pri programovaní v čo najširšom merítke využívať dostupné funkcie API dostupné v príslušnej verzii OS Symbian. Neustále pamätať na problémy spojené so silne obmedzeným systémovými prostriedkami (hlavne operačná pamäť).

Aplikácia v podstate disponuje funkciami popísanými v kapitole 6.2. Za zmienku ešte stojí popis ako bolo plánované dosiahnuť funkcie náhľadu obrázkov a aké by to malo dopady na súčasnú podobu aplikácie.

Na obrázku 7.1 je vidieť využiteľný typ zoznamu pre naše potreby. Žltá podfarbená plocha predstavuje jednu položku zoznamu. V našej aplikácii by zobrazovala jeden súbor alebo adresár. Ak by to bol obrázok, tak by k nemu zobrazovala náhľad a pomocou textového poľa by bol zobrazený názov a dodatočné informácie (veľkosť, dátum vytvorenia, atď.). Po vybraní nejakého obrázku by bolo možné ho otvoriť. Ak by položka predstavovala adresár, tak by bolo možné do neho vstúpiť a zobraziť jeho obsah. Týmto by odpadol hlavne spôsob otvárania obrázkov v súčasnej aplikácii. Bolo by možné otvoriť ľubovoľný zvolený obrázok. Teraz je to obmedzené na konkrétne štyri obrázky. Ale na predvedenie určitej funkčnosti to plne postačuje. Už spomínaná funkcia *ScaleL()* by bola použitá na vytváranie náhľadov. Po otvorení obrázku by sa pomocou tejto funkcie zmenšilo rozlíšenie na približne 24 x 24 pixelov a takýto malý obrázok by sa potom uložil ako obrázok pre vytváranie rýchleho náhľadu.

Prípadné vyriešenie ukladania obrázkov napr. vo formáte JPEG by malo za následok hlavne zrýchlenie zápisu súboru do pamäte.

Takže hlavne kvôli týmto dvom problémom nieje výsledná aplikácia taká elegantná ako by sa od nej očakávalo. Pri testovaní výkonnosti som dospel k záveru, že aplikácia dokáže otvoriť obrázok do približného rozlíšenia 1500 x 1500 obrazových bodov. Otvorenie takto veľkých obrázkov silne závisí na veľkosti voľnej operačnej pamäte. Otvorenie trvá rádovo sekundy a tento čas je porovnateľný s inými aplikáciami. Ostatné operácie odohrávajúce sa už v operačnej pamäti sú pekne svižné. Výhodou tejto aplikácie je aj fakt, že dokáže obrázky otvoriť zo súboru a súčasne ich dokáže a odфотографovať. Užívateľovi teda odpadá prácné prepínanie sa medzi viacerými aplikáciami.

8 Záver

V rámci tejto práce vznikol pohľad na históriu vývoja operačného systému Symbian a jeho použitie na poli mobilných telefónov. V dnešnej dobe sa ich zastúpenie na trhu stále zvyšuje. Prispieva k tomu najmä jednoduchá a praktická možnosť rozšírenia mobilného telefónu o nové funkcie.

Z programátorského hľadiska vznikol hlavne prehľad vývojových nástrojov dostupných na poli Smartphoneov založených na operačnom systéme Symbian. Konkrétne sa zaoberá návrhom a vývojom aplikácií pre mobilný telefón Siemens SX1 a jemu podobné. Pri výbere vývojového prostredia padla voľba na MS Visual C++ 6.0. v kombinácii s SDK, ktoré dodáva výrobca telefónu. Sú tu preto popísané jednotlivé kroky pri vývoji novej aplikácie za pomoci prostriedkov tohto vývojového prostredia a možnosti emulácie vyvíjanej aplikácie na integrovanom emulátore, ktorý verne zobrazuje telefón, pre ktorý je aplikácia vyvíjaná.

Nasleduje návrh konkrétnej aplikácie, ktorá má byť ukážkou možností práce s obrazovými dátami na mobilných telefónoch s operačným systémom Symbian. Navrhované vlastnosti aplikácie kladú dôraz na to, aby bola aplikácia dostatočne funkčne zaujímavá a hlavne výkonnostne robustná. Návrh bol realizovaný aj s ohľadom na existujúce aplikácie. Čiastočne sa nimi inšpiruje a hlavne sa snaží eliminovať ich negatívne vlastnosti. Hlavné problémy s výkonnosťou.

Pri realizácii navrhutej aplikácie sa vyskytli určité problémy. Na začiatku bol problém so získaním akejkoľvek literatúry. Postupne som získal knihu o Symbiane v angličtine a nakoniec aj v češtine. Obe knihy (viď zoznam literatúry [13] a [14]) sú ale skôr referenčné manuály ako obsiahle príručky s príkladmi. Hlavne anglická verzia je dosť strohá a neobsahuje skoro žiadne príklady použitia. Česká verzia zase používa už verziu Symbianu 7.0, takže niektoré uvedené konštrukcie sú pre mňa v podstate nepoužiteľné.

Výsledná aplikácia preto nemá úplne všetky navrhované funkcie, ale napriek tomu je plne funkčná a poskytuje základné operácie na prácu s obrázkami. Spája možnosť fotografovania aj otváranie obrázkov zo súboru. Podrobnejší popis dosiahnutých výsledkov a porovnanie s inými aplikáciami je v predchádzajúcej kapitole.

Pretože realizovaná aplikácia zostala čiastočne v nedokončenom stave je možné v tejto práci pokračovať a ďalej ju zdokonaľovať. Je možné dopracovať navrhnuté funkcie a doplniť iné. Prípadným záujemcom o vývoj aplikácií pre mobilné telefóny s operačným systémom Symbian ale silne doporučujem zvážiť pre akú verziu OS Symbian chcú aplikáciu vyvíjať. Ako minimálnu verziu by bolo vhodné zvoliť verziu 7.0. Tá už totiž má obsiahlejšie API a hlavne pre ňu existuje aj slušná literatúra a aj na rôznych webových stránkach a diskusiách sa vyskytuje neporovnateľne viac ukážok a užitočných príkladov. Pri zvolení staršej verzie môžu totiž naraziť na podobné problémy ako som

mal ja. A nikoho nepoteší, keď desiatky hodín strávi pracnými a často zbytočnými pokusmi, zisťujúc správny postup pre vytvorenie požadovanej funkcie. Táto verzia už má prepracovaný aj prístup k integrovanej kamere. Umožňuje preto celkom efektívnu prácu s videom a jednoduchšiu prácu s obrázkami. Hlavne ich konverziu medzi jednotlivými formátmi. Takže sa ponúkajú projekty typu rozpoznávanie obrazu a podobne.

Literatúra

- [1] Jaroslav Kováč: Symbian – král na trhu smartphonov, URL:
<http://www.mobilmania.sk/Mobilnetelefony/Ar.asp?ARI=5975&CHID=1&EXPS=&EXPA=>
(máj 2007)
- [2] Ivo Mareček: Symbian: kompletní přehled všech smartphonů u nás i ve světě, URL:
<http://www.mobilmania.cz/kapesni/AR.asp?ARI=110900> (máj 2007)
- [3] mobilmania.cz
<http://www.mobilmania.cz> (máj 2007)
- [4] Cisco systems
<http://www.cisco.com> (máj 2007)
- [5] Barva v počítačové grafice
<http://cs.felk.cvut.cz/~chrasa1/apgbarvy/barvy.html> (máj 2007)
- [6] Karel Pechanec: Vývojové nástroje pro Symbian, URL:
<http://www.rdc.cz/index.php?jazyk=0&sid=3&main=sekce&stav=clanek&cid=217> (máj 2007)
- [7] NewLC.com
<http://www.newlc.com/Nokia-Series-60-SDK.html> (máj 2007)
- [8] Karel Pechanec: HelloWorld aplikace v C++ pro Nokia 7650, URL:
<http://www.rdc.cz/index.php?jazyk=0&sid=3&main=sekce&stav=clanek&cid=207> (máj 2007)
- [9] Dokumentácia k SDK v1.2 pre MS Visual C++
- [10] Symbian.com
<http://www.symbian.com/> (máj 2007)
- [11] My-Siemes.com
<http://www.my-siemens.com/> (máj 2007)
- [12] Digimanie.cz
<http://www.digimanie.cz> (máj 2007)
- [13] Leigh Edwards, Richard Barker: Developing Series 60 Applications, 2004
- [14] Richard Harrison: Programujeme aplikace pro Symbian OS v jazyce C++, Brno, 2006

Príloha 1

Súbor bmpmanip.pkg

```
; Bmpmanip.pkg
;

;Language - standard language definitions
&EN

; standard sis file header
#{ "bmpmanip" }, (0x10005B80), 1, 0, 0

; Supports Series 60 v 0.9
(0x101F6F88), 0, 0, 0, { "Series60ProductID" }

;
"\Symbian\6.1\Series60\epoc32\release\armi\urel\bmpmanip.APP"-
"C:\system\apps\bmpmanip\bmpmanip.app"
"\Symbian\6.1\Series60\epoc32\release\armi\urel\bmpmanip.rsc"-
"C:\system\apps\bmpmanip\bmpmanip.rsc"
"\Symbian\devel\bmpmanip\gifs\image1.gif"- "C:\system\apps\bmpmanip\image1.gif"
"\Symbian\devel\bmpmanip\gifs\image1.bmp"- "C:\system\apps\bmpmanip\image1.bmp"
"\Symbian\devel\bmpmanip\gifs\image1.jpg"- "C:\system\apps\bmpmanip\image1.jpg"
"\Symbian\devel\bmpmanip\gifs\image1.png"- "C:\system\apps\bmpmanip\image1.png"
```

Niektóre funkcje z pliku bmpmanipappview.cpp:

```
//Open picture file
```

```
void CBmpmanipAppView::OpenPicture(const TInt PictureType)
```

```
{
```

```
    switch (PictureType)
```

```
    {
```

```
    case 0: // open bmp
```

```
        //Start an asynchronous process to open BMP picture
```

```
        iConverter->OpenL(KBmpOpenFileName);
```

```
        break;
```

```
    case 1: // open png
```

```
        //Start an asynchronous process to open Png picture
```

```
        iConverter->OpenL(KPngOpenFileName);
```

```
        break;
```

```
    case 2: // open jpg
```

```
        //Start an asynchronous process to open Jpg picture
```

```
        iConverter->OpenL(KJpgOpenFileName);
```

```
        break;
```

```
    case 3: // open gif
```

```
        //Start an asynchronous process to open Jpg picture
```

```
        iConverter->OpenL(KGifOpenFileName);
```

```
        break;
```

```
    default:
```

```
        //Start an asynchronous process to open BMP picture
```

```
        iConverter->OpenL(KBmpOpenFileName);
```

```
    }
```

```
}
```

```
//Draw this applications view to the screen
```

```
void CBmpmanipAppView::Draw(const TRect& /*aRect*/) const
```

```
{
```

```
    // clear the screen
```

```
    CWindowGc& gc = SystemGc();
```

```
    gc.Clear(Rect());
```

```
//if the bitmap is not currently being processed in any way
```

```
if (iConvertState == EConvertStateReady || iConvertState == EConvertStateSaving)
```

```

        {
            gc.DrawBitmap(Rect(), iBitmap, iBitmap->SizeInPixels());
        }
    }

void CBmpmanipAppView::RotateImageClockwiseL(TImageRotateAngle aAngle)
{
    if (iConvertState != EConvertStateReady)
    {
        //Inform the UI that the gif file is still being loaded
        User::Leave(KErrInUse);
    }
    switch (aAngle)
    {
    case E90Degrees:
        //Start an asynchronous process to rotate the bitmap
        iRotator-
        >RotateL(*this,*iBitmap,CMdaBitmapRotator::ERotation90DegreesClockwise);
        iConvertState = EConvertStateRotating;
        break;
    case E180Degrees:
        //Start an asynchronous process to rotate the bitmap
        iRotator-
        >RotateL(*this,*iBitmap,CMdaBitmapRotator::ERotation180DegreesClockwise);
        iConvertState = EConvertStateRotating;
        break;
    case E270Degrees:
        //Start an asynchronous process to rotate the bitmap
        iRotator-
        >RotateL(*this,*iBitmap,CMdaBitmapRotator::ERotation270DegreesClockwise);
        iConvertState = EConvertStateRotating;
        break;
    default:
        ASSERT(FALSE);
    }
}

```

```

void CBmpmanipAppView::SaveL()
{
    if (iConvertState != EConvertStateReady)
    {
        return;
    }

    //Start an asynchronous process to create a file for the bitmap
    //Note this does not save the actual bitmap - this is done once the file has been created
    iFileSaver->CreateL(KBmpSaveFileName,&iClipFormat,&iCodec,&iCodec);
    iConvertState = EConvertStateSaving;
}

void CBmpmanipAppView::Save1BppL()
{
    if (iConvertState != EConvertStateReady)
    {
        return;
    }

    //Start an asynchronous process to create a file for the bitmap
    //Note this does not save the actual bitmap - this is done once the file has been created
    iFileSaver->CreateL(KBmpSaveFileName,&iClipFormat,&i1BppCodec,&i1BppCodec);
    iConvertState = EConvertStateSaving;
}

////////////////////////////////////

void CBmpmanipAppView::ListL()
{
    iEikonEnv->InfoWinL(_L("Tato funkce není implementována"), _L(":o"));
}

```

```
TBool CBmpmanipAppView::IsCameraReady()
```

```
{  
    ASSERT(iCameraManager != NULL);  
    return iCameraManager->IsReady();  
}
```

```
void CBmpmanipAppView::Snap()
```

```
{  
    ASSERT(iCameraManager != NULL);  
    iCameraManager->Snap(*iBitmap);  
    iConvertState = EConvertStateReady;  
}
```


Príloha 2

Súčasťou tejto dokumentácie je aj CD. Toto CD má nasledujúcu štruktúru:



- Adresár **dokumentacia** obsahuje dokumentáciu k projektu vo formáte .doc a .pdf.
- Adresár **aplikacie** obsahuje zdrojové súbory ukážkovej aplikácie helloworld a vypracovanej aplikácie bmpmanip
- Adresár **vyvojove nástroje** obsahuje SDK v1.2 S60 pre MS Visual C++, inštaláciu ActivePerl-5.6.1.638, inštaláciu j2re-1_3_1_18 a súbor skin-sx1.zip, ktorý obsahuje potrebné súbory na zmenu emulátoru tak, aby vyzeral ako telefón Siemens SX1.