

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## POKROČILÝ PROHLÍŽEČ OBRÁZKŮ PRO WINDOWS MOBILE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

TOMÁŠ LEXMAUL

BRNO 2012



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# **POKROČILÝ PROHLÍŽEČ OBRÁZKŮ** **PRO WINDOWS MOBILE**

ADVANCED IMAGE VIEWER FOR WINDOWS MOBILE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ LEXMAUL**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ZBYNĚK KŘIVKA, Ph.D.**

BRNO 2012

## Abstrakt

V této práci je popsán vývoj pokročilého prohlížeče obrázků pro systém Windows Mobile. Program nabízí konfigurovatelnost uživatelského rozhraní a ovládání pomocí gest. Doplňuje již existující prohlížeče a přináší moderní způsob ovládání i pro starší systém. Aplikace je implementována v programovacím jazyce C#.

## Abstract

This bachelor thesis describes the development of the Advanced Image Viewer for Windows Mobile. The program offers a configurable user interface and control with gestures. It complements the existing browsers and offers a modern way to control for older systems. This application is implemented in C# programming language.

## Klíčová slova

Prohlížeč obrázků, Windows Mobile, .NETCF, C#

## Keywords

Picture viewer, Windows Mobile, .NETCF, C#

## Citace

Tomáš Lexmaul: Pokročilý prohlížeč obrázků  
pro Windows Mobile, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Pokročilý prohlížeč obrázků pro Windows Mobile

## Prohlášení

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

.....  
Tomáš Lexmaul  
13. května 2012

## Poděkování

Tímto bych chtěl poděkovat panu Zbyňku Křivkovi za vedení práce a odbornou pomoc.

© Tomáš Lexmaul, 2012.

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

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Technologie</b>	<b>4</b>
2.1 Personal Digital Assistant	4
2.2 Windows Mobile	4
2.2.1 Historie Windows Mobile	5
2.2.2 Windows Mobile 2003	5
2.2.3 Windows Mobile 5.0	5
2.2.4 Windows Mobile 6	5
2.2.5 Windows Phone 7	6
2.3 Vývoj aplikací pro WM	6
2.3.1 .NET Compact Framework	7
2.3.2 Jazyk C#	7
2.3.3 Základní stavební prvky	7
2.4 Nejpoužívanější grafické formáty	9
2.4.1 JPG	9
2.4.2 BMP	9
2.4.3 PNG	9
2.4.4 GIF	10
2.5 Vybrané knihovny pro načítání obrázků a EXIF informací	10
2.5.1 OPENNETCF Smart Device Framework	10
2.5.2 Accusoft ImageGear	11
2.5.3 LEADTOOLS Mobile Imaging Developer Toolkit	11
2.5.4 ExifLib	11
2.5.5 GSensorSDK	11
<b>3 Analýza a návrh aplikace</b>	<b>12</b>
3.1 Funkční požadavky	12
3.1.1 Existující řešení	12
3.1.2 Základní požadavky na aplikaci	14
3.2 Návrh uživatelského rozhraní aplikace	15
3.2.1 Nastavení	16
3.2.2 Gesta pro ovládání aplikace	17
<b>4 Implementace</b>	<b>19</b>
4.1 Implementované třídy	19
4.2 Použité funkce ze Smart Device Framework	20
4.3 Problém načítání obrázků a velikosti paměti	20

4.4	Rozpoznávání a učení gest . . . . .	22
4.4.1	Algoritmus rozpoznávání gest . . . . .	22
4.5	Načítání EXIF informací . . . . .	22
4.6	Načítání a ukládání nastavení . . . . .	23
4.7	Dialogy pro otevírání souborů a složek . . . . .	23
4.8	Implementace podpory alb . . . . .	23
4.9	Posouvání obrázku . . . . .	24
<b>5</b>	<b>Testování aplikace</b>	<b>28</b>
5.1	Použitá zařízení . . . . .	28
5.2	Navržené testy . . . . .	28
5.3	Vyhodnocení testů . . . . .	29
<b>6</b>	<b>Závěr</b>	<b>32</b>
6.1	Porovnání s existujícími prohlížeči . . . . .	32
6.2	Přínos práce . . . . .	32
6.3	Navrhovaná vylepšení . . . . .	33
6.4	Budoucnost programu . . . . .	33
<b>A</b>	<b>Tok obrazovek</b>	<b>36</b>

# Kapitola 1

## Úvod

Prohlížení obrázků je v dnešní době jeden z nejčastěji prováděných úkonů na mobilních zařízeních. Většina mobilních telefonů obsahuje poměrně kvalitní fotoaparáty, zvláště potom chytré telefony s vlastním operačním systémem. Trhu dominují operační systémy Android od společnosti Google, iOS od společnosti Apple a Windows Phone od společnosti Microsoft. Operační systém Windows Mobile od společnosti Microsoft (dále jen WM) je sice dnes již zastarávající, nicméně jej stále mnoho uživatelů používá. Pro modernější systémy existuje nespočet různě kvalitních prohlížečů obrázků využívajících funkčnosti telefonů.

Pro vývojáře WM není tak atraktivní platformou jako dříve. Jednak se již další zařízení s tímto systémem nevyrábějí, ale také neexistuje oficiální distribuční kanál programů, jako je tomu u novějších systémů. Zařízení s WM jsou často omezeny výkonem a velikostí paměti, vývojář se potýká s problémem velmi rozdílného výkonu jednotlivých telefonů. Systém WM také obsahuje různé limitace, zvláště potom omezení paměti pro jednotlivé procesy a nepodporuje moderní funkce, jako je ovládání pomocí více dotykových gest a využití pohybového senzoru.

Pro systém WM existuje velké množství prohlížečů, které jsou různé kvality. Smyslem této bakalářské práce je vytvoření prohlížeče obrázků s podporou formátů JPG, BMP, GIF a PNG, který bude obsahovat konfigurovatelné rozhraní, zaměřené na ovládání gesty. Nabídne tak uživateli možnost používat pokročilé funkce i na starších zařízeních.

Úvodní kapitola je zaměřena na popis vybraných technologií. Je zde rozebrán operační systém Windows Mobile, dále vývoj aplikací pro mobilní zařízení, nepoužívanější grafické formáty a vybrané knihovny pro načítání obrázků. Zabývá se také knihovnou pro načítání informací ve formátu Extensible Image File Format (dále jen EXIF) z obrázků a knihovnou pro podporu pohybového a polohového senzoru telefonů značky HTC.

V další kapitole je čtenář seznámen s existujícími řešeními, funkčními požadavky a návrhem aplikace.

V kapitole 4 je popsána implementace aplikace a rozebrány použité algoritmy.

Kapitola 5 se zabývá testováním prohlížeče a vyhodnocením testů.

V závěru je podáno srovnání s existujícími prohlížeči, zhodnocení přínosu práce a popsána navrhovaná vylepšení.

## Kapitola 2

# Technologie

Kapitola se zabývá platformou WM, její historií a vývojem aplikací pro tento systém. Rozebírá základní stavební prvky a představuje nejpoužívanější grafické formáty. Dále popisuje knihovny pro načítání obrázků, pro načítání EXIF informací a pro práci s polohovým snímačem telefonu.

### 2.1 Personal Digital Assistant

*Personal Digital Assistant* (dále jen PDA), česky *osobní digitální pomocník* či *malý kapesní počítač*, je zařízení malých rozměrů vybavené dotykovým displejem. Ovládá se pomocí *stylusu*<sup>1</sup> nebo prstů. Je schopno komunikovat s okolím pomocí bezdrátových technologií nebo kabelů. Umožňuje výměnu dat a synchronizaci s PC. Obsahuje vlastní procesor, paměť a operační systém. Hardwarová specifikace pro PDA s WM se nazývá Pocket PC.

### 2.2 Windows Mobile

Windows Mobile je operační systém od firmy Microsoft určený pro vyspělá mobilní zařízení. Je odvozen z Windows CE, který je určen pro vestavěná zařízení.

Společnost Microsoft se snažila, aby uživatelské rozhraní tohoto systému bylo podobné Windows pro PC. Proto byly vytvořeny aplikace jako Office mobile, Internet Explorer pro WM, Media player pro WM či Outlook Mobile. Spolupráce s Windows pro PC je také na velmi dobré úrovni. Nabízí synchronizaci zpráv, kontaktů a kalendáře. Umožňuje také snadné sdílení internetového připojení.

WM podporuje tyto tři základní typy mobilních zařízení:

- Chytré telefony – Smartphone
- Komunikátory – Pocket PC Phone Edition
- Kapesní počítače – Pocket PC

Chytré telefony neobsahují dotykový displej. Komunikátory i kapesní počítače dotykový displej obsahují, přičemž komunikátory mají ve výbavě i telefonní modul.

---

<sup>1</sup>Podlouhlá tyčinka pro ovládání dotykového displeje. Umožňuje přesnější ovládání než prsty.



### 2.2.1 Historie Windows Mobile

Historie Windows Mobile sahá až do roku 2000, kdy byl představen systém Pocket PC2000. Z něj dále vycházel systém Pocket PC 2002. Oba tyto systémy byly založeny na Windows CE 3.0. Většího rozšíření se WM dočkalo až s příchodem SmartPhone 2002. Tento systém byl jako první určen pro chytré telefony. Nejznámější telefon s tímto systémem byl Motorola MPx200, který se prosadil díky nízké ceně. Informace a obrázky jsou převzaty z [9] a [10].

### 2.2.2 Windows Mobile 2003

Tento systém (obr. č. 2.1) je založen na Windows CE .NET 4.2. Hlavními změnami oproti dřívějším edicím byly zvýšení stability, vylepšené možnosti komunikace pomocí Bluetooth a Wi-Fi a zavedení podpory pro práci s multimédií. Dále byla přidána podpora VGA rozlišení displeje, možnost změny orientace displeje a podpora pro netradiční poměry stran.



Obrázek 2.1: Úvodní obrazovka Windows Mobile 2003

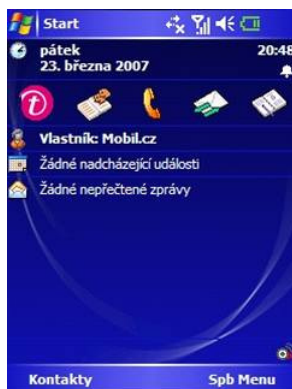
### 2.2.3 Windows Mobile 5.0

Windows Mobile 5.0 (obr. č. 2.2) byl představen v roce 2005. Vychází z Windows CE .NET 5.2 a přinesl s sebou určité inovativní prvky. Hlavní změnou byl odlišný přístup k paměti zařízení. V předchozích verzích sloužila paměť RAM jako datové úložiště a systémová paměť zároveň. Ve WM 5.0 slouží výhradně pro běh aplikací a všechna ostatní data se ukládají do FlashROM. Dále byla přidána podpora USB 2.0, vylepšená funkčnost synchronizace, nový kancelářský balík Office a podpora pro vysouvací QWERTY klávesnici.

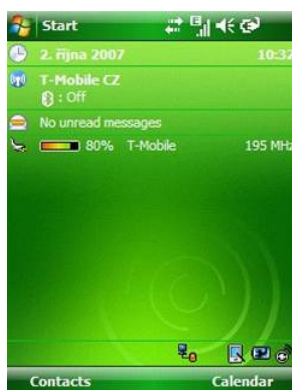
### 2.2.4 Windows Mobile 6

Windows mobile 6 (obr. č. 2.3) byl představen na začátku roku 2007. Uživatelské rozhraní bylo přepracováno způsobem umožňujícím ovládání bez použití stylusu. Byla vylepšena odezva systému, přidána možnost šifrovat i data uložená na paměťové kartě, možnost ovládání vzdálené plochy, možnost uspořádat SMS zprávy do vláken a také byl přidán VOIP klient pro internetovou telefonii.

Z WM 6 vychází Windows Mobile 6.5, který přináší mimo jiné vylepšené ovládání a podporu gest. Na obrázku č. 2.4 je zobrazena úvodní obrazovka. Uživatelské rozhraní je určené hlavně pro ovládání prsty.



Obrázek 2.2: Úvodní obrazovka Windows Mobile 5.0



Obrázek 2.3: Úvodní obrazovka Windows Mobile 6

### 2.2.5 Windows Phone 7

Kvůli sílícímu tlaku ze strany konkurence a klesajícímu podílu WM na trhu byla společnost Microsoft nucena vyvinout zcela nový operační systém pro mobilní telefony. Tím je právě Windows Phone 7 (obr. č. 2.5). Uživatelské rozhraní bylo navrženo velmi rozdílným způsobem než u WM. Základním stavebním prvkem jsou dlaždice, do kterých si může uživatel napojit libovolné aplikace. Další prvek rozcestníky (Hubs) jsou ovládací centra mající za úkol pomoci uživateli v nejběžnějších operacích jako správa a prohlížení kontaktů, prohlížení fotek a nakupování aplikací. Mobilní telefony pro Windows Phone 7 musí splňovat určité požadavky jako multidotykový kapacitní displej s poměrem stran 5:3 a minimálním rozlišením WVGA. Bohužel chybí zpětná kompatibilita aplikací, což je způsobeno velmi rozdílnou koncepcí systému. [5]

## 2.3 Vývoj aplikací pro WM

Společnost Microsoft poskytuje pro vývoj aplikací vývojové prostředí Visual Studio. Pro mobilní zařízení je třeba verze minimálně Professional, nižší verze vývoj mobilních aplikací nepodporují. V základní instalaci obsahuje Visual Studio 2008 knihovny pouze pro Windows Mobile 5.0, proto pro aplikace určené pro Windows Mobile 6 je potřeba doinstalovat



Obrázek 2.4: Úvodní obrazovka Windows Mobile 6.5

Windows Mobile 6 SDK nebo novější. Tato sada vývojových nástrojů obsahuje dokumentaci, ukázkové programy se zdrojovými kódy, hlavičky a knihovny, obrazy jednotlivých verzí operačního systému pro emulátor a nástroje pro přeložení aplikace pro cílovou platformu. Windows Mobile 6 SDK je dostupný ve třech verzích [8]:

- Standard – určeno pro chytré telefony Smartphone
- Classic – kapesní počítače Pocket PC
- Professional – komunikátory Pocket PC Phone Edition

### 2.3.1 .NET Compact Framework

.NET Compact Framework (dále .NETCF) je podmnožina plného .NET Framework určeného pro systém Windows. Umožňuje vývoj aplikací určených pro Windows Mobile. Visual Studio 2008 podporuje verze 2.0 a 3.5. .NETCF používá některé z knihoven plného .Net Framework, ale obsahuje také knihovny vytvořené přímo pro mobilní zařízení. Kvůli malé paměti těchto zařízení jsou knihovny omezené funkcími a velikostí. Vývoj uživatelského rozhraní v .NETCF je založen na formulářích z klasických Windows. Zde se používá myš, která je u mobilních zařízení nahrazena stylusem. Kvůli tomu není rozhraní příliš vhodné pro ovládání prsty. Existují specializované knihovny, které tento problém řeší.

### 2.3.2 Jazyk C#

Aplikace pro WM je možné psát v různých jazycích. Jedním z nich je C#. Jedná se o moderní, mnohoúčelový, objektově orientovaný programovací jazyk. Poskytuje nástroje pro bezpečnější tvoření aplikací jako hlídání hranic polí, detekci používání neinicializovaných proměnných a automatický *garbage collector*. Tento jazyk je navržen pro různé platformy – Windows, Windows Mobile a XBOX 360.

### 2.3.3 Základní stavební prvky

Při koncipování tohoto oddílu jsem vycházel z knihy *Microsoft Mobile Development Handbook* [16]. Pro vytvoření programu můžeme využít jeden a více projektů, přičemž jeden z nich je spustitelný. Jednotlivé projekty mohou být určeny pouze pro vytvoření dynamicky linkované knihovny (DLL) nebo pro vytvoření samotného programu. Spustitelný projekt



Obrázek 2.5: Úvodní obrazovka Windows Phone 7

obsahuje soubory s informacemi o projektu, soubory, které se budou přikládat, dále soubor s kódem pro spuštění programu a soubory s kódem k jednotlivým formulářům, popř. třídám. Obsahuje také seznam použitých knihoven. Pokud se jedná o běžnou knihovnu dostupnou ve WM, nemusí se k programu přikládat. V opačném případě je nutné přiložit i tuto knihovnu.

Formuláře jsou hlavním stavebním prvkem programu pro mobilní zařízení. Na tyto formuláře se potom nabalují další ovládací prvky. Na hlavní formulář lze napojit další formuláře, jak ilustruje následující kód.

```
Form MyForm = new Form();  
MyForm.Parent = this;  
MyForm.Show();
```

Nejdříve se vytvoří nový formulář, poté se nastaví vlastnictví na současný formulář a posléze se okno vykreslí. Pokud použijeme metodu `Show`, současný formulář není pozastaven a běží v pozadí. Pokud chceme formulář pozastavit, použijeme metodu `ShowDialog`.

Vytváření nových formulářů obsahuje jistou režii, proto pokud by se měla okna měnit často, lze využít pouze jeden formulář a měnit jeho obsah. Jednotlivé prvky lze buďto přesunout mimo viditelnou plochu, nebo lze využít parametru `Visible` a zneviditelnit je.

Nejběžnějším prvek je `Menu` pro vytvoření nabídky, dále prvek `Label` pro zobrazení textu, `Button` jako tlačítko a další. Pro seskupení prvků můžeme využít i prvek `Panel`.

Pro uspořádání lze též s výhodou použít prvek `TabControl`. Další ovládací prvky jsou potom rozčleněny do záložek. Nevýhodou může být, že nelze uživateli zabránit v přechodu na jinou záložku.

V prohlížeči obrázků je zřejmě nejvíce využívaný prvek `PictureBox`. Tento prvek je určen pro vykreslování obrázků na obrazovku. `.NETCF` obsahuje knihovnu `System.Drawing`, která

mimo jiné obsahuje metody pro načítání obrázku. Pro načtení obrázku lze tedy použít následující kód.

```
Bitmap bmp = new Bitmap(obrazek.pripona)
```

Tento způsob načítání má ovšem jednu velkou nevýhodu, obrázek se načte v plné velikosti a zabírá tedy velké množství paměti. Proto není tato metoda vhodná pro vytvářený prohlížeč.

Pro otevírání souborů .NETCF nabízí prvek *OpenFileDialog*. Pro ukládání souborů potom *SaveFileDialog*. WM ve verzi 6 a vyšší nabízí také *SelectPictureDialog* pro otevírání obrázků. Dialog pro otevírání složky není implementován, jak je zmíněno dále.

Pro testování programu lze využít buď reálného zařízení, nebo vestavěný emulátor. Je potřeba nejdříve stáhnout soubory obsahující jednotlivé verze WM.

WM používá instalační balíčky CAB (Composite UI Application Block). Umožňuje programátorovi nastavit instalaci programu, připojit potřebné soubory (jako jsou např. ikony) a dále připojit potřebné knihovny.

## 2.4 Nejpoužívanější grafické formáty

Grafických formátů existuje mnoho. Liší se v mnoha ohledech podle použití. V profesionálních fotoaparátech se využívá např. formátu RAW, kde data ze záznamového čipu nejsou nijak upravována a jsou přímo uložena. Výhodou je tedy zachování nejvyšší možné kvality. Nevýhodou je pochopitelně vysoká paměťová náročnost. Další bezztrátový formát je TIFF (Tagged Image File Format). Podporuje maximální barevnou hloubku 32 bitů a umožňuje uložit metainformace o obrázku ve formátu EXIF.

Zadání této práce požaduje podporu 4 formátů: JPG, BMP, PNG a GIF. Tyto formáty si tedy představíme. Následující informace jsou čerpány z [11] a [12].

### 2.4.1 JPG

JPEG (Joint Photographic Experts Group) je ztrátový grafický formát. Byl standardizován v roce 1994. Označení JPEG zahrnuje jednak formát souboru, kterým je ukládán a také grafický kodek JPEG/JFIF (JPEG File Interchange Format). Tento kodek slouží k převodu a komprimování obrazových dat. V dnešní době je tento formát pravděpodobně nejpoužívanější. Vyniká zejména velmi malým objemem dat při podpoře plného počtu barev (24 bitů) a zachování vyhovující kvality. Používá se především u kompaktních fotoaparátů a na internetu. Další výhodou je možnost uložení metainformací ve formátu EXIF. Nevýhodou může být, že nepodporuje průhlednost.

### 2.4.2 BMP

Tento formát využívá nejjednodušší způsob uložení grafické informace. Jednotlivé pixely jsou zde zaznamenány přesně tak, jak jdou za sebou. Pro každý pixel je k dispozici barevná informace až do hloubky 24 bitů. Výhodou tohoto formátu je bezztrátové uložení obrázku.

### 2.4.3 PNG

PNG (Portable Network Graphics) byl vyvinut jako náhrada za placený GIF. Používá bezztrátovou kompresi pomocí algoritmu DEFLATE. Podporuje barevnou hloubku až do 32

bitů, 24 bitů je určeno pro uložení obrázku a 8 bitů je tzv. Alpha kanál, ve kterém se zaznamenávají stíny a podobné grafické efekty. Tento formát se používá zejména k ukládání webové grafiky a exportů z vektorových grafických formátů, jako náhrada za zastaralý a patentovaný GIF.

#### 2.4.4 GIF

GIF (Graphics Interchange Format) je patentovaný a placený grafický formát pro bezztrátové ukládání obrázků. Nevýhodou je barevná hloubka pouze 8 bitů (256 barev). I přesto je formát hojně používaný především pro ukládání obrázků u webové grafiky. Je dobře použitelný i na ukládání vektorové grafiky v rastrové podobě. Výhodou je také podpora animací.

## 2.5 Vybrané knihovny pro načítání obrázků a EXIF informací

Pro načítání obrázků poskytuje .NETCF základní nástroje. Tyto nástroje mají omezenou funkčnost a neumožňují např. načítání velkých obrázků a jejich následné zmenšení bez plného rozbalení v paměti. Na internetu jsem vyhledal několik dostupných alternativ. Podarilo se mi najít pouze jednu sadu nástrojů pro práci z obrázky, která podporuje všechny zadané formáty, je v určité edici zdarma a je určena přímo pro Windows Mobile. Dále jsem vybral dvě placené alternativy.

### 2.5.1 OPENNETCF Smart Device Framework

Jedná se o sadu nástrojů určenou pro vývoj Windows Mobile aplikací. Tato sada je vyvíjena společností OpenNETCF Consulting. Je dostupná ve třech edicích [14].

- Comunity
- Standard
- Professional

Edice Comunity je dostupná zdarma, zbývající edice jsou placené. Edice Comunity poskytuje stejnou funkčnost, ale obsahuje určitá omezení. Např. nejsou dostupné zdrojové kódy, chybí dokumentace integrovaných tříd, neobsahuje šablony s předkonfigurovanými referencemi a není poskytována podpora.

Smart Device Framework (dále SDF) obsahuje také knihovnu pro načítání obrázků a metadat ve formátu EXIF. Tato knihovna dále poskytuje nástroje pro práci s obrázky jako otočení, převrácení, vytvoření náhledu, změna barevné palety či změna přiblížení. Podporuje načítání značně velkých obrázků a jejich následné zmenšení bez využití velkého množství paměti. Je tedy vhodná i pro práci s fotografiemi vysokého rozlišení.

Knihovna podporuje načítání všech zadaných formátů obrázků . tj. JPG, PNG, GIF a BMP.

SDF je dostupný pod licencí *Shared Source License*, která je k dispozici na stránkách společnosti [13]. Umožňuje šířit zkompileované knihovny společně s aplikací, která je uvolněna pod licencí, jež nevyžaduje dodání zdrojových kódů.

### 2.5.2 Accusoft ImageGear

Jedná se o specializovaný SDK pro práci s obrázky, vyvíjený společností Accusoft. Podporuje načítání velkého množství různých formátů obrázků. Dále nabízí nástroje pro zpracování jako zaostření, detekci hran, redukci barevného prostoru, úpravu kontrastu, rotaci a další. Podporuje také pokročilé úpravy jako vylepšení rozlišitelnosti pro OCR zpracování. Pro prohlížení obrázků nabízí nástroje jako posouvání, přibližování a vyhlazování.

SDK je dostupné ve zkušební verzi na stránkách společnosti [1]. Cena tohoto řešení je nyní 2999\$ a možnosti licencování jsou dostupné na [2].

### 2.5.3 LEADTOOLS Mobile Imaging Developer Toolkit

Jedná se o sadu nástrojů pro práci s obrázky. Podporuje načítání množství různých formátů. Poskytuje nástroje pro zpracování obrázků jako změnu barevného rozlišení, převedení na černobílý obrázek, změnu velikosti, otočení, oříznutí, převrácení a kompresi. Pro zobrazování nabízí např. nástroj lupa, posouvání, vyhlazování, úpravu jasu, kontrastu a gamma korekce.

Mobile Imaging Developer Toolkit je dostupný ve zkušební verzi na stránkách společnosti [7]. Cena tohoto řešení je nyní 595\$. Licence je dostupná na [6].

### 2.5.4 ExifLib

EXIF (Exchangeable image file format) je specifikace pro formát metadat vkládaných do obrázků. Tyto informace umožňují uchování údajů například o:

- Znače a modelu fotoaparátu
- Datu a čase pořízení snímku
- Nastavení fotoaparátu
- Místě pořízení snímku
- Náhled snímku

Knihovna ExifLib umožňuje načítání EXIF informací z obrázků. Knihovna je původně určená pro desktop, ale po malém upravení se může použít i v mobilním telefonu.[15]

### 2.5.5 GSensorSDK

Jedná se o knihovnu umožňující používat pohybový a polohový senzor v telefonech HTC. Při změně orientace telefonu vyvolá událost, umožňující programu na ni zareagovat a přizpůsobit se nové orientaci. [4]

## Kapitola 3

# Analýza a návrh aplikace

Požadavky, které jsou uvedeny v zadání práce bylo potřeba analyzovat. Vhodnou technikou je nejdříve prostudovat vlastnosti již implementovaných prohlížečů, které jsou na trhu, zjistit jejich výhody a nedostatky a podle získaných poznatků se pokusit navrhnout nový prohlížeč.

Kapitola se věnuje analýze a zhodnocení existujících řešení prohlížečů obrázků, dále stanovuje požadavky na funkčnost implementované aplikace. Věnuje se návrhu uživatelského rozhraní a gest pro ovládání prohlížeče.

### 3.1 Funkční požadavky

Cílem práce je navrhnout konfigurovatelný prohlížeč obrázků použitelný v mobilním telefonu. Prohlížeč by měl být uživatelsky přívětivý, ale musí poskytovat pokročilé funkce. Prostudováním vlastností existujících řešení jsem získal poznatky, které jsem využil při specifikaci požadavků na funkčnost vytvářené aplikace.

#### 3.1.1 Existující řešení

Windows Mobile patří k velmi rozšířeným systémům, přestože je dnes již vytlačován jinými systémy. Pro ilustraci současného stavu prohlížečů obrázků pro tento systém jsem vybral několik aplikací volně stažitelných na internetu, popř. již integrovaných v systému. Většina z nich je zdarma, pouze Resco Photo Manager je placený.

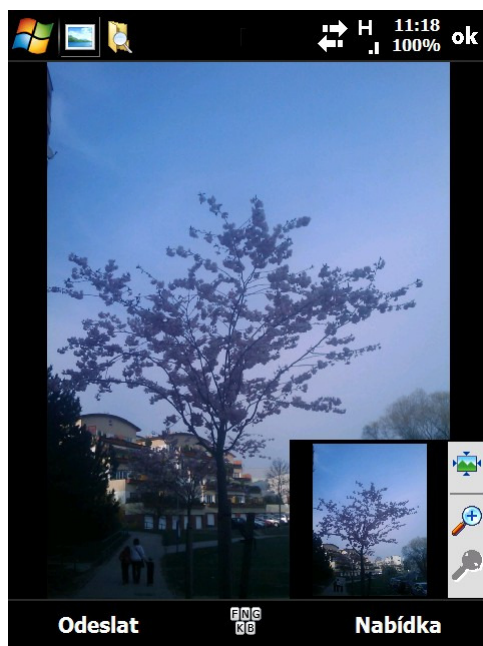
#### Vestavěný prohlížeč obrázků ve Windows Mobile

Vestavěný prohlížeč obrázků ve Windows Mobile (obr. č. 3.1(a)) disponuje základními vlastnostmi jako jsou: lupa, posouvání obrázku, prezentace, poslání obrázku a jednoduchá editace. Jako jediný dokáže zobrazit i značně velké obrázky v plném přiblížení. Ve verzi pro Windows Mobile 6.1 nepodporuje žádná gesta, ve verzi pro WM 6.5 podporuje posun na další/předchozí obrázek. Nabízí také základní konfiguraci.

#### Vestavěný HTC prohlížeč

Vestavěný HTC prohlížeč disponuje modernějším vzhledem (obr.č. 3.1(b)) a uživatelskou přívětivostí. Podporuje plynulý přechod na další obrázek. Jeho funkcionality je velmi omezená. Podporuje pouze samotné prohlížení a přiblížení do určité úrovně, dále režim pre-





(a) Vestavěný prohlížeč ve WM



(b) Vestavěný HTC prohlížeč

Obrázek 3.1: Vybrané prohlížeče obrázků

zentace a odeslání obrázku. Z gest podporuje pouze posunutí na další/předchozí obrázek. Nenabízí žádné možnosti konfigurace.

### Picture Viewer 2.3

Picture Viewer 2.3 (obr. č. 3.2(a)) podporuje prohlížení, otočení obrázku o 90° a jako jediný podporuje i další gesta, avšak jejich použití není příliš intuitivní.

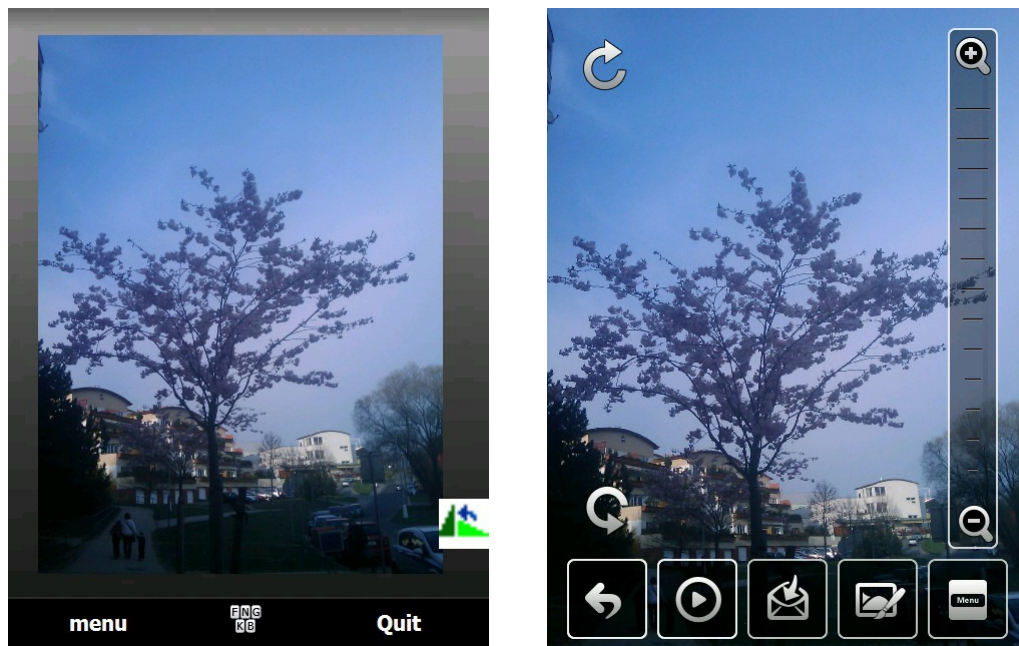
### Resco Photo Manager

Resco photo manager (obr. č. 3.2(b)) je jako jediný zástupce placený program. Jedná se o velmi pokročilý prohlížeč s mnoha funkcemi, vynikajícím moderním vzhledem a rychlostí. Podporuje všechny základní funkce jako předchozí prohlížeče, navíc implementuje i automatické otočení obrazovky podle polohy telefonu. Nabízí i jednoduchou editaci obrázku nebo převod formátů. Umožňuje také svou konfiguraci.

### Dava Picture Viewer

Dava Picture Viewer (obr. č. 3.3), stejně jako ostatní prohlížeče podporuje všechny základní funkce. Nabízí navíc podporu vytváření alb. Umožňuje také načítání základních EXIF informací.

Každý z těchto prohlížečů je zaměřený trochu jiným směrem. Zatímco vestavěné prohlížeče nabízejí základní funkce a doplňují uživatelské rozhraní systému, ostatní prohlížeče se snaží



(a) Picture Viewer 2.3

(b) Resco Photo Manager

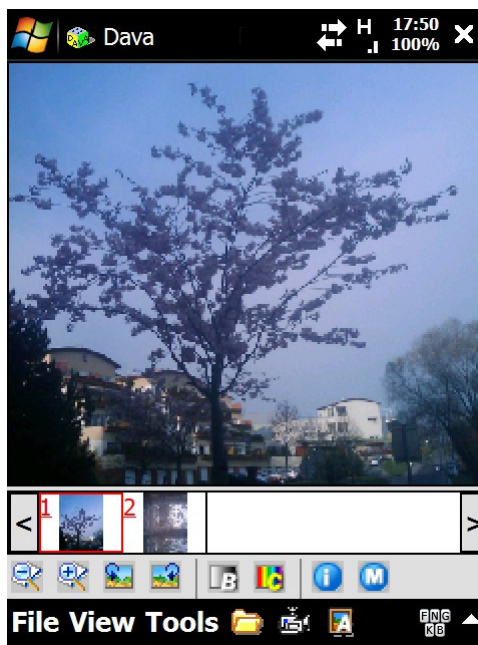
Obrázek 3.2: Vybrané prohlížeče obrázků

zaujmout jiným způsobem. Dava Picture Viewer umožňuje tvorbu alb, kterou jsem se rozhodl využít i v mojí aplikaci. Picture Viewer podporuje ovládání pomocí gest, která jsou ale poměrně nezdařilá. Resco Photo Manager je velmi kvalitní program, který podporuje dokonce i polohový senzor telefonu HTC. Nepodporuje ale pokročilé ovládání pomocí gest. Navrhovaná aplikace tedy využije vlastností zkoumaných programů a pokusí se je implementovat v použitelné podobě.

### 3.1.2 Základní požadavky na aplikaci

Po předchozí analýze existujících řešení prohlížečů byly vytvořeny následující požadavky na implementovanou aplikaci:

- Aplikace je funkční na zařízeních s dotykovým displejem a s operačním systémem WM ve verzi 6 a vyšší.
- Aplikace podporuje načítání obrázků ve formátech JPG, PNG, GIF a BMP. Hlavní obrazovka obsahuje prostor pro zobrazení samotného obrázku, prostor pro zobrazení nástrojů a alba. Zobrazené nástroje se mohou zvolit. V nastavení lze nástroje a album schovat.
- Aplikace umožňuje tvorbu alb, jejich ukládání a modifikaci.
- Aplikace umožňuje ovládání pomocí uživatelských gest. Ke každému nástroji je přiřazeno určité gesto, které lze v nastavení predefinovat.
- Manipulace s obrázkem je možná pomocí gest, ale také pomocí položek v menu.
- Prohlížeč umožňuje otevírat obrázky jednotlivě, ale i po celých složkách.



Obrázek 3.3: Dava Picture Viewer

- Aplikace podporuje minimálně rozlišení 320x240 a 640x480.
- Aplikace podporuje polohový senzor a upravuje obrazovku podle natočení telefonu.
- Aplikace umí zobrazit EXIF informace z načteného obrázku.
- Je dostupný režim prezentace s možností konfigurace, především rychlosti a směru posunu a také umístění nástrojů.
- Prohlížeč umožňuje i načítání větších obrázků, jako jsou fotografie z telefonu s možností přiblížení do určité úrovně.

## 3.2 Návrh uživatelského rozhraní aplikace

Po prozkoumání možností knihoven .NETCF a OPENNETCF jsem navrhl funkčnost aplikace. Využil jsem přitom volně dostupné ikony z [17]. Hlavní obrazovka je na obrázku č. 3.4. Skládá se ze tří částí a panelu se dvěma menu. V horní části je vykreslen prohlížený obrázek. Pod obrázkem je panel s náhledy dalších obrázků v albu. Následuje panel s jednotlivými nástroji. Tento panel obsahuje nástroje pro přiblížení/oddálení, zarovnání na obrazovku a maximalizaci, nástroj ruka pro posun přiblíženého obrázku, nástroj gesta pro režim gest, nástroje pro otočení a prezentaci, dále potom nástroje pro otevření, přidání nebo odebrání obrázku a pro otevření nebo přidání adresáře. Zobrazeny jsou pouze nástroje, které se vejdou na obrazovku. Jednotlivé položky je možné přidávat a odebírat v nastavení. Panel s náhledem obrázků i panel s nástroji je možné odstranit v nastavení.

Na obrázku č. 3.5(a) je zobrazeno hlavní *Menu*. Položka *Otevřít obrázek* uzavře aktuální otevřené album a otevře jeden obrázek. Položka *Přidat obrázek* zařadí nově otevřený obrázek do právě otevřeného alba, opačně potom položka *Odebrat obrázek*. Obdobně položky *Otevřít*



Obrázek 3.4: Hlavní obrazovka aplikace

*složku* a *Přidat složku*, které pracují s množstvím obrázků v jednotlivých složkách. Položka *Otevřít album* otevře uložené album ve formátu XML, položka *Uložit album jako...* nabídne jeho uložení. Položka *Nastavení* otevře nabídku nastavení aplikace.

Na obrázku č. 3.5(b) je ukázáno menu *Nástroje*. Položka *Informace* umožňuje zobrazení metainformací obrázku ve formátu EXIF. Položka *Zarovnání* umožňuje nastavit výchozí režim zarovnání obrázku na šířku/výšku, automaticky, či bez zarovnání. Položka *Přiblížení* obsahuje možnost procentuálního přiblížení – 100%, 75%, 50% a 25%. Položka *Prezentace* umožňuje spuštění prezentace.

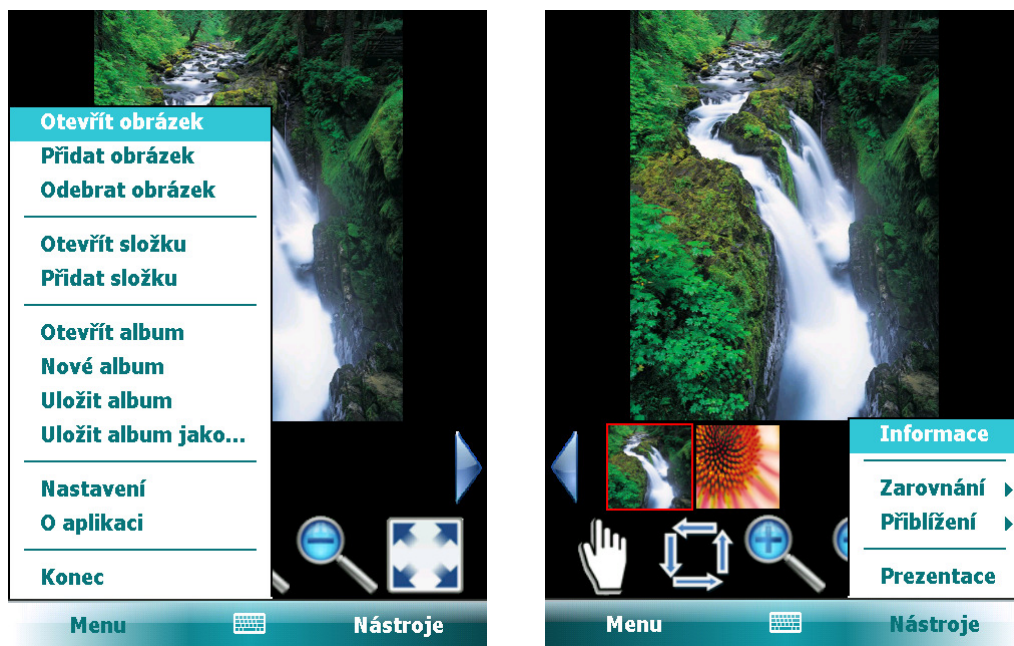
### 3.2.1 Nastavení

Obrazovka s nastavením je rozdělena do několika záložek:

- Vzhled
- Zobrazení
- Nástroje
- Prezentace
- Gesta
- Dialogy

V záložce *Vzhled* je možno nastavit zobrazení/skrytí jednotlivých položek hlavní obrazovky – horní systémové menu, panel náhledů obrázků a panel nástrojů.

Záložka *Zobrazení* umožňuje nastavení automatického otáčení obrázků podle EXIF informací, otáčení obrazovky podle polohy telefonu a implicitní zarovnání obrázku – na šířku/výšku obrazovky, originální velikost a automatické zarovnání na obrazovku.



(a) Hlavní menu

(b) Menu Nástroje

Obrázek 3.5: Menu prohlížeče

V záložce *Nástroje* je možno nastavit jednotlivé položky, které se budou zobrazovat na panelu nástrojů na hlavní obrazovce.

V záložce *Prezentace* je možno nastavit směr prezentace – dopředu/dozadu. Dále možnosti umístění panelu a jeho zobrazení – zobrazit/skrýt/automaticky skrývat. Následuje možnost nastavení umístění panelu nástrojů na strany obrazovky.

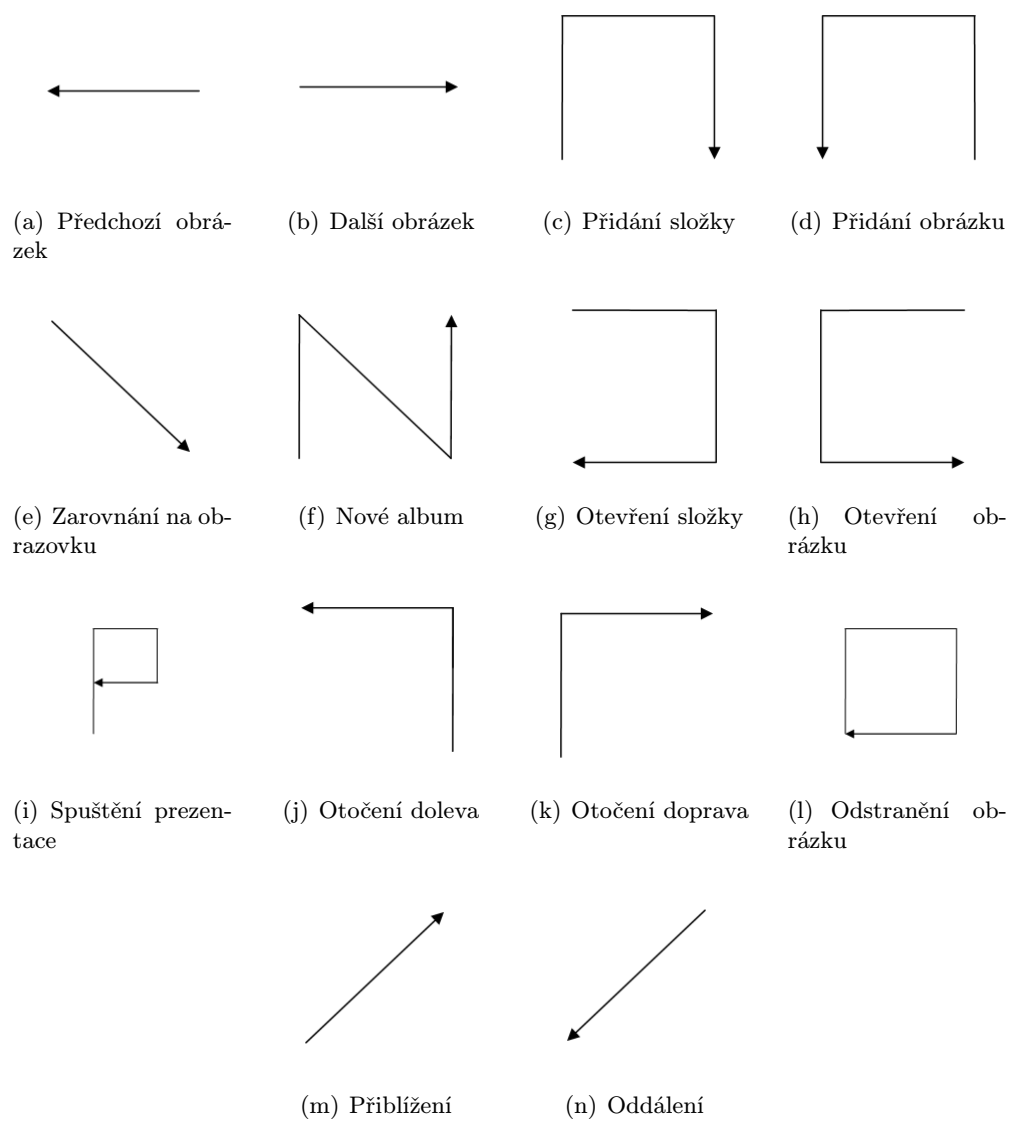
V záložce *Gesta* je možno nastavit samotné povolení gest. Dále je možno pro každé gesto zobrazit detaily, kde je ukázáno aktuální gesto a původní gesto. Zde je možnost nadefinování vlastního gesta nebo obnovení původního gesta.

Záložka *Dialogy* umožňuje zvolit, který dialog se použije při otevírání obrázku. K dispozici jsou dialogy s náhledy, systémový dialog a vlastní stromový dialog.

### 3.2.2 Gesta pro ovládání aplikace

Standardní .NET Compact Framework bohužel neobsahuje nástroje pro práci s gesty. Pouze Windows mobile 6.5 podporuje rozpoznávání základních gest, která jsou pro potřeby programu nedostatečná. Proto byly vytvořeny nástroje pro rozpoznávání a učení gest. Na následujících obrázcích jsou ukázány původní podoby gest. Systém rozpoznávání podporuje pouze gesta složená z úseček.

Gesta je možno používat, pouze když je prohlížeč v režimu gesta. V opačném případě je prohlížeč v režimu ruka, sloužícím pro posun přiblíženého obrázku. Mezi těmito režimy je možno přepínat jednak nástroji a také dvojklikem. Ke každému nástroji z panelu nástrojů je přiřazeno právě jedno gesto. Na obrázku č. 3.2.2 jsou zobrazeny návrhy pro implicitní ovládání prohlížeče.



Obrázek 3.6: Gesta k ovládání prohlížeče

## Kapitola 4

# Implementace

Kapitola popisuje jednotlivé problémy implementace a rozebírá použité algoritmy.

Aplikace je implementována v programovacím jazyce C# za použití .NETCF 3.5 pro platformu Windows Mobile. Aplikace podporuje telefony s operačním systémem WM 6.0 a vyšší. Pro využití polohového senzoru je potřeba používat telefon značky HTC.

### 4.1 Implementované třídy

Pro demonstraci implementovaných tříd byl vytvořen diagram (obr. č. 4.1). V diagramu jsou zahrnuty pouze vybrané metody a členské proměnné.

Hlavní obrazovka je implementována třídou `GestureImageViewer` odvozenou ze třídy `Form`. Obsahuje jednak prvek `PictureBox` pro zobrazení samotného obrázku, dále potom ostatní prvky `PictureBox` pro zobrazení nástrojů a náhledů alba. Dále obsahuje `Menu` s ovládacími tlačítky. Při každé změně velikosti obrazovky, zejména při prvním načtení a změně orientace se vyvolává událost `Resize`, která se ošetřuje metodou `OnResize`. Program rozmisťuje prvky dynamicky, proto bylo nutné tuto metodu přepsat.

```
protected override void OnResize(EventArgs e)
```

Tato metoda zjistí, jestli se mají zobrazovat panely alba a nástrojů, dále zjistí, kolik se jich na obrazovku vejde, a podle toho je umístí na správné pozice. Také vyhradí prostor pro vykreslení obrázku. Prohlížeč se tak stal nezávislým na rozlišení displeje a jeho orientaci.

Pro načtení obrázku byla implementována metoda `LoadImage`.

```
private void LoadImage(string path)
```

Metoda přijímá pouze cestu k obrázku. Podle nastavení zarovnání volá další metody, které se postarají o vykreslení v příslušné velikosti.

Jakmile je obrázek načten v paměti, o jeho vykreslení se stará metoda `DrawImageFromBitmap`. Podle aktuální pozice v obrázku a podle velikosti prvku `PictureBox` vybere část načtené bitmapy a zajistí její vykreslení. `Backbuffer` prvku `PictureBox` s vyrovnávací pamětí je pojmenován `ToPaint`. Z něj je vytvořen objekt třídy `Graphics` určený pro vykreslování. Nejdříve je vyplněn černou barvou, poté se do něj zapíše část z bitmapy `LoadedBitmap`. Posléze se prvek `PictureBox` obnoví a zobrazí zapsaný výřez, jak je ukázáno v následujícím kódu.

```

using (Graphics g = Graphics.FromImage(pictureBox1.ToPaint))
{
    g.Clear(Color.Black);
    g.DrawImage(LoadedBitmap, WhereToDrawX,
                WhereToDrawY, Rec, GraphicsUnit.Pixel);
    pictureBox1.Refresh();
}

```

Obrazovka s nastaveními je implementována třídou `SettingsForm`. Pro lepší uspořádání voleb nastavení je využito prvku `TabControl`, který umožňuje rozdělení do záložek. Objekt třídy `SettingsForm` při inicializaci přijímá strukturu `MySettings`, která obsahuje všechny informace o nastaveních. Podle načtených údajů nastaví všechny ovládací prvky nastavení. Po deaktivaci okna opět vrací strukturu `MySettings` se změněným nastavením.

Obrazovka s náhledem stávajícího gesta a možností definice nového je implementována třídou `GestureLearn`. Obsahuje 3 záložky: náhled současného gesta, náhled původního gesta s možností obnovení a záložku pro definici nového gesta.

Obrazovky pro otevření souboru a složky jsou implementovány třídami `OpenFile` a `OpenFolder`. Využívají prvku `TreeView` pro zobrazení stromové struktury adresáře telefonu.

Obrazovka pro zobrazení údajů o obrázku je implementována třídou `EXIF`. Používá prvek `ListView` pro vypsání údajů. Obsahuje metodu pro získání a zpracování informací z načteného obrázku.

Zpracování gest je implementováno třídou `MyGestureRecognizer`. Obsahuje metody pro rozpoznání gesta ze sesbíraných bodů a také jeho porovnání s jiným gestem.

Třída `MySettings` obsahuje proměnné pro uchování všech potřebných nastavení a metody pro jejich načtení a uložení do souboru.

Nástroje pro práci s albem jsou implementovány třídou `Album`. Obsahuje také metody pro uložení a načtení ze souboru.

`PictureBox` s podporou technologie `Double buffering` je implementován třídou `BufferedPictureBox`.

## 4.2 Použité funkce ze Smart Device Framework

SDF nabízí celou řadu knihoven pro zjednodušení tvorby programů pro systém Windows mobile. Rozhodl jsem se využít knihovnu pro práci s obrázky `Drawing.dll` a dále knihovnu pro ukládání a načítání nastavení `AppSettings.dll`.

První zmiňovaná knihovna umožňuje načítání základních informací o obrázcích jako jsou šířka, výška a bitová hloubka. Dále umožňuje samotné načtení obrázku podle zvolené velikosti, včetně vytvoření náhledu obrázku. Poskytuje také nástroj pro rotaci obrázku či jeho převrácení.

Knihovna pro načítání nastavení poskytuje nástroje pro ukládání potřebných informací do souboru ve formátu XML.

## 4.3 Problém načítání obrázků a velikosti paměti

Mobilní telefony se systémem Windows mobile obvykle bývají limitovány velikostí operační paměti. Zařízení s WM 6 a vyšším mívají desítky až stovky megabajtů, ale reálně je použitelných mnohem méně. Systém WM bohužel limituje paměť každého procesu na 32MB. Reálně lze ovšem použít maximálně 18-20MB.



S použitím komprese lze uložit i značně velké obrázky, které ovšem po dekompresi mohou zabírat i desítky megabajtů v paměti. Proto bylo nutné zvolit jistý kompromis mezi velikostí použité paměti a velikostí zobrazeného obrázku. Pozorováním a prozkoušením bylo zjištěno, že lze přidělit přibližně 6MB spojitěho kusu paměti. Proto byla tato hranice zvolena pro maximální velikost rozbaleného obrázku. S použitím knihovny ze SDF lze ovšem načítat obrázky značné velikosti (více než 10MPx), které jsou knihovnou zmenšeny tak, aby zabíraly méně než zmíněných 6MB. Pro 24 bitovou hloubku obrázku jsou to 2MPx.

Další problém s limitovanou pamětí nastává při rotaci obrázku. Pokud je obrázek čtvercový, je možno ho otáčet o 90° bez alokace další paměti. V opačném případě je potřeba použít další kus paměti stejné velikosti. SDF poskytuje nástroj pro pravoúhlé rotace obrázku.

```
LoadedBitmap = ImageUtils.Rotate(bmp,
    OpenNETCF.Drawing.RotationAngle.Clockwise90);
```

Tento nástroj byl použit pro menší obrázky, protože je velmi rychlý. Při otáčení většího obrázku ale vyvolává výjimku `OutOfMemoryException`, tedy nedostatek paměti. Proto bylo implementováno vlastní řešení. WM umožňuje alokaci paměti v globálním prostoru pomocí funkce `Marshal.AllocHGlobal(MemorySize)`;

Bohužel ani v tomto případě se nepodařilo přidělit potřebné množství paměti v celku. Proto byl vytvořen algoritmus, který dokáže rotovat obrázek o 90° a pracovat přitom s několika částmi paměti. Algoritmus nejdříve přesune rotovaný obrázek do částí paměti, jak je znázorněno na obrázku č. 4.2. Potom je starý obrázek uvolněn pomocí volání `Dispose()` a následuje zkopírování otočeného obrázku zpátky a uvolnění dočasné paměti. Celý proces je výpočetně náročný, na druhou stranu umožňuje rotovat i velké obrázky.

Pro přístup k jednotlivým pixelům lze použít metody `GetPixel` a `SetPixel`. Tento způsob je ale velmi pomalý. Jedná se o řízený kód (managed code). Pro přímý přístup do paměti v jazyce C# je potřeba použít klíčové slovo `unsafe`. Zavolání metody `LockBits` umožní programátorovi přistupovat k pixelům přes ukazatele, tj. přímo do paměti. Následující kód demonstruje iteraci přes jednotlivé pixely v obrázku.

```
BitmapData bmd = pic.LockBits(new Rectangle(0, 0, 20, 20),
    System.Drawing.Imaging.ImageLockMode.ReadWrite,
    pic.PixelFormat);
int PixelSize=3;

for(int y=0; y<bmd.Height; y++)
{
    byte* row=(byte *)bmd.Scan0+(y*bmd.Stride);
    for(int x=0; x<bmd.Width; x++)
    {
        row[x*PixelSize]=255;
    }
}
```

Volání `LockBits` vrací objekt `BitmapData`, který obsahuje informace o adrese začátku přistupovaných dat v paměti (`Scan0`). Položka `Stride` uchovává hodnotu délky řádku v bytech i se zarovnáním. `Width` a `Height` obsahují šířku a výšku zamknutého prostoru. Kód

znázorňuje nastavení červené složky všech pixelů na hodnotu 255. Pracuje se se 24 bitovým obrázkem (hodnota `PixelFormatSize` je nastavena na 3 bajty). Po ukončení se musí zavolat metoda `UnlockBits` pro následné odemčení.

## 4.4 Rozpoznávání a učení gest

Standardní .NET Compact Framework bohužel neobsahuje nástroje pro práci s gesty. Pouze WM 6.5 podporuje rozpoznávání základních gest, která jsou pro potřeby programu nedostatečná. Proto byly vytvořeny nástroje pro rozpoznávání a učení gest. Pro získávání dat z obrazovky je využito událostí `OnMouseDown`, při které se zachytí první potenciální bod gesta, dále `OnMouseMove`, kde se posbírají všechny body pohybu a `OnMouseUp`, kde se zachytí poslední bod gesta a započne se zpracování a vyhodnocení. Při zadávání gesta na obrazovku se bohužel nesejmou všechny body po kterých se pohybovalo, proto bylo nutné vyplnit mezery mezi body úsečkami, aby bylo gesto spojitě.

### 4.4.1 Algoritmus rozpoznávání gest

Pro rozpoznání gest jsem navrhl jednoduchý algoritmus. Tento algoritmus dostane pro zpracování posloupnost nasbíraných bodů z obrazovky. Zjistí se nejmenší a největší souřadnice, podle nich se vytvoří čtverec tak, aby krajní body gesta byly jeho hranicí. Tento čtverec se dále rozdělí do devíti oblastí, jak je vidět na obrázku č.4.3. Podle sesbíraných bodů se určuje, které oblasti byly protnuty a v jakém pořadí. Pokud by bylo gesto složeno jen z vertikálních a horizontálních tahů, tento systém by pracoval bez problému. Pokud ovšem zahrneme i šikmé tahy, je potřeba algoritmus vylepšit.

Uvažujme gesto – úsečku z levého spodního rohu do pravého horního rohu. Ideálně by úsečka měla protnout pouze tři oblasti. Levý spodní, prostřední a pravý horní čtverec. V reálném případě ovšem částečně protne i okolní čtverce, i když jen malou délkou. Proto bylo zavedeno pravidlo, které tento problém řeší. Před zpracováním gesta se vypočítá průměrná hodnota počtu bodů v jednotlivých oblastech. Pokud při zpracování počet bodů v oblasti je menší než 30% této průměrné hodnoty, oblast se nezapočítává. Výhodou tohoto algoritmu je nezávislost na velikosti a pozici gesta.

Třída `MyGestureRecognizer` obsahuje všechny potřebné metody pro zpracování gest.

## 4.5 Načítání EXIF informací

OpenNETCF obsahuje nástroje pro načítání EXIF informací, bohužel ale není dostačující. Informace se zobrazují pouze v číselné podobě a většina z nich se ani nenačítá. Proto byla zvolena knihovna `ExifLib`, která se po malé úpravě dá použít i ve WM. Knihovna umožňuje čtení informací včetně polohy zařízení při pořízení snímku. Dále podporuje například i čtení GPS souřadnic, na kterých byl obrázek pořízen.

Moderní fotoaparáty často disponují polohovými senzory. Dokážou tedy zjistit, v jaké poloze byl snímek pořízen. Jelikož je rotace obrázku výpočetně náročná operace, fotoaparát obrázek nerotuje přímo, ale opatří jej značkou EXIF Orientation Flag. V prohlížeči, kde rychlost není tak kritická, se podle této značky může obrázek otočit.

Tato značka může nabývat osmi hodnot. V tabulce č. 4.5 je znázorněno, jak bude obrázek orientován při různých hodnotách. Hodnoty 2,4,5,7 jsou málo používané, protože se jedná o převrácený obraz. V implementovaném prohlížeči byla tedy přidána podpora pouze

Hodnota EXIF značky orientace	Řádek č. 0 je umístěn	Sloupec č. 0 je umístěn
1	Nahoře	Vlevo
2	Nahoře	Vpravo
3	Dole	Vpravo
4	Dole	Vlevo
5	Vlevo	Nahoře
6	Vpravo	Nahoře
7	Vpravo	Dole
8	Vlevo	Dole

Tabulka 4.1: Orientace obrázku podle značky EXIF

pro hodnoty 1,3,6 a 8. Na obrázku č. 4.4 je znázorněna poloha zařízení při focení a dále požadovaný směr otočení v prohlížeči.

## 4.6 Načítání a ukládání nastavení

Pro načítání a ukládání nastavení jsem použil knihovnu *AppSettings* z OpenNETCF. Potřebné informace se transformují do číselné nebo textové podoby. Knihovna je potom uloží ve formátu XML do souboru. Načítání probíhá obdobně.

## 4.7 Dialogy pro otevírání souborů a složek

Windows Mobile 6 poskytuje 2 základní dialogy použitelné pro otevření obrázku. `OpenFileDialog` a `SelectPictureDialog`.

`SelectPictureDialog` (obr. č. 4.5(a)) zobrazuje náhledy obrázků. U větších obrázků ovšem může jejich zobrazení trvat delší dobu. Také nepodporuje načítání formátu png.

`OpenFileDialog` (obr.č. 4.5(b)) je pro načítání souborů použitelný. Liší se podle jednotlivých verzí systému. Ve WM 6.0 nepodporuje zobrazení stromové struktury adresáře, proto je navigace obtížnější. Ve WM 6.1 je již dialog mnohem lépe navržený.

Pro otevření složky bohužel žádný dialog neexistuje. Proto byly vytvořeny dva vlastní dialogy. Pro otevření souboru `OpenFile` (obr. č. 4.6(a)) a pro otevření složky `OpenFolder` (obr. č. 4.6(b)). Dialogy jsou velmi podobné. Využil jsem prvku `TreeView` pro zobrazení nalezených souborů a složek.

## 4.8 Implementace podpory alb

Prohlížeč obrázků pracuje s alby v základu a je jen na uživateli, zda se jich rozhodne využít. Na hlavní obrazovce je zobrazen panel s náhledy obrázků v albu. Do alba je možno přidávat obrázky po jednom nebo i po složkách, popřípadě obrázků odstranit. Uložením alba se uchovávají informace o umístění obrázků a jejich posloupnosti. Využívá se přitom dříve zmíněné knihovny *AppSettings.dll* a ukládá se ve formátu XML.

## 4.9 Posouvání obrázku

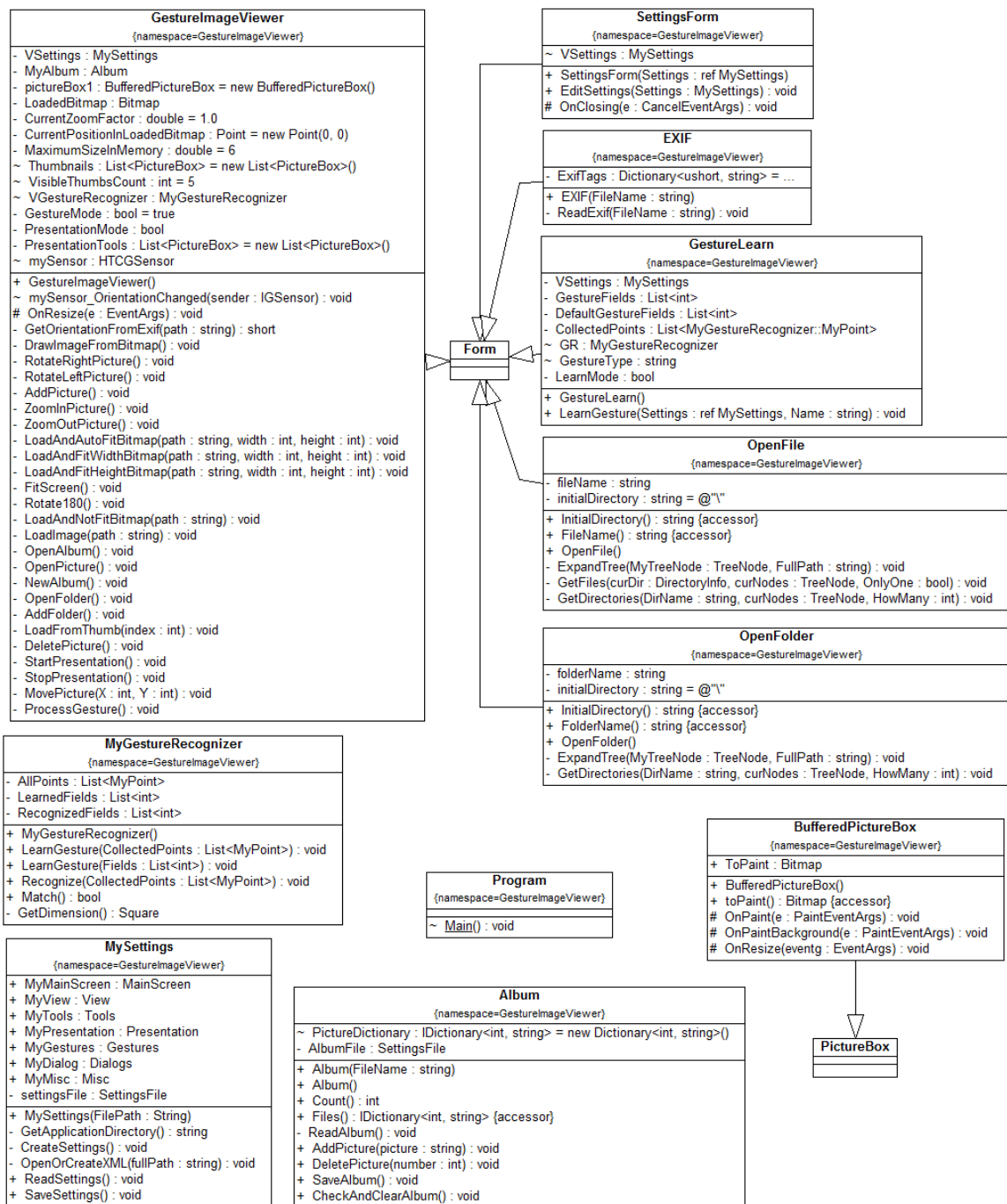
Pro posun v obrázku větším, než je obrazovka, bylo původně zamýšleno využít nástrojů dostupných ve WM 6.5. V tom případě by ale bylo posouvání ve starších verzích nedostupné. Navíc bylo zjištěno, že při použití těchto nástrojů se události `OnMouseDown` a `OnMouseMove` chovají nekorektně. Událost `OnMouseDown` se vyvolá až po určitém posunu na obrazovce, tento systém se tedy stává nepoužitelným pro rozpoznávání gest. Proto byl posun obrázku implementován ručně. Do paměti je uložen celý obrázek v daném přiblížení a vykresluje se pouze příslušná část.

Pro zabránění trhání při posouvání obrázku bylo využito techniky *Double buffering* [16]. Bez použití této techniky se obrázek vykresluje na obrazovku po částech. Při pohybu obrázku tedy dochází ke zkreslení. Řešením je zavedení vyrovnávací paměti tzv. *Back buffer*, do které se nejdříve zkopíruje obraz k vykreslení a teprve potom se z této vyrovnávací paměti vykreslí na obrazovku. Pohyb je potom plynulý.

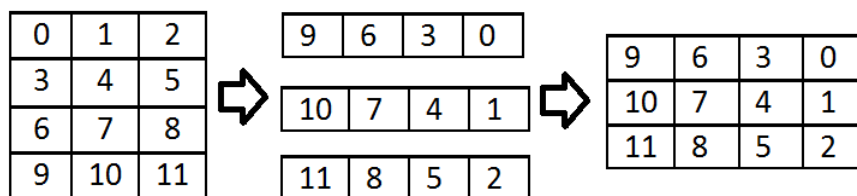
Pro použití v prohlížeči byla implementována třída `BufferedPictureBox` vycházející z třídy `PictureBox`, která obsahuje vyrovnávací paměť `ToPaint`. Do této paměti se nejdříve zkopíruje vykreslovaná oblast. Při události `OnPaint` se potom obsah vykreslí na obrazovku, jak značí následující kód. Pokud je vyrovnávací paměť prázdná, vykreslí se černá obrazovka.

```
protected override void OnPaint(PaintEventArgs e)
{
    if (ToPaint != null)
    {
        e.Graphics.DrawImage(ToPaint, 0, 0);
    }
    else
        e.Graphics.Clear(this.BackColor);
}
```

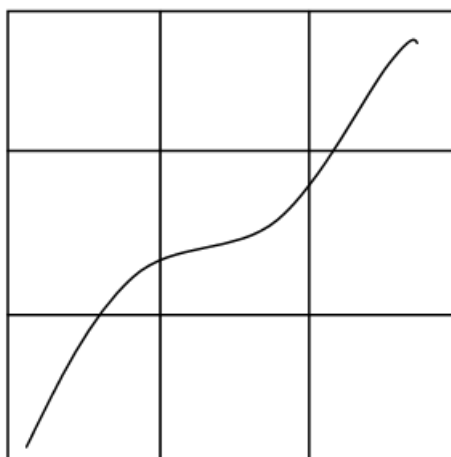
Dále se musí přepsat obsluha události `OnPaintBackground`, která vykresluje pozadí, a docházelo by tak k viditelnému trhání.



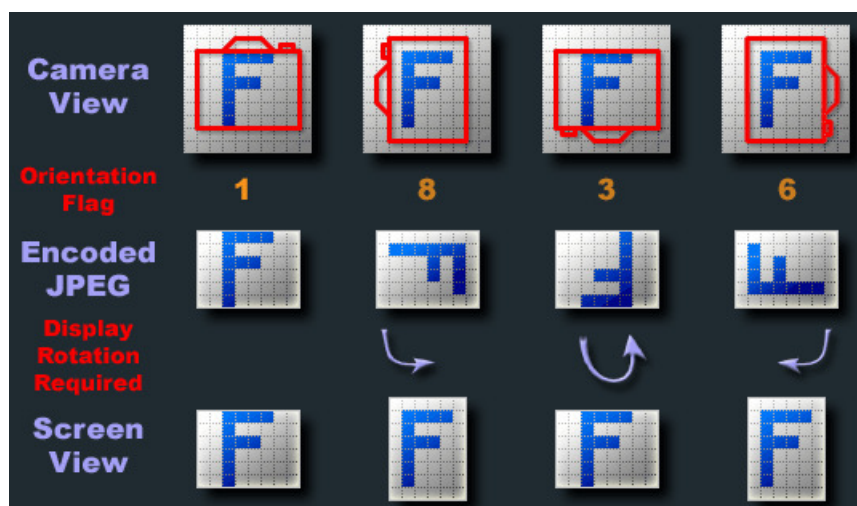
Obrázek 4.1: Diagram implementovaných tříd



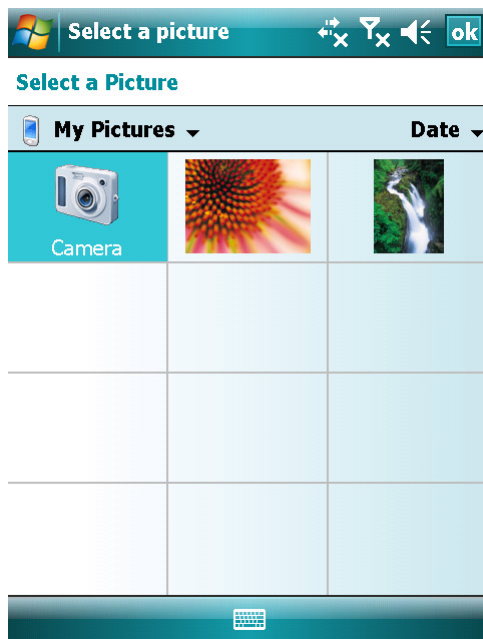
Obrázek 4.2: Rotace obrázků po pixelech s využitím více částí paměti



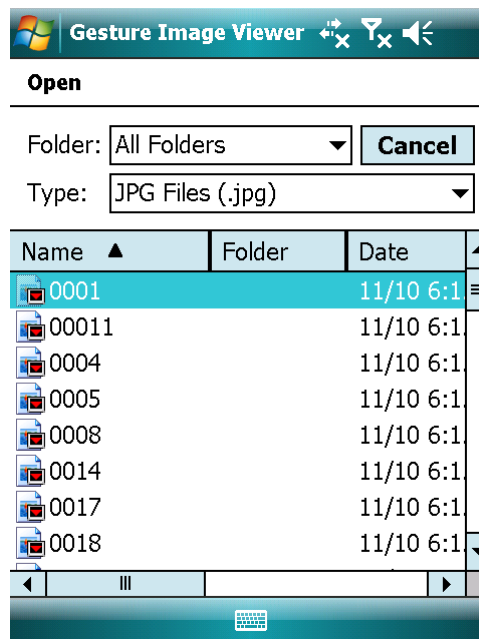
Obrázek 4.3: Ukázka reálného gesta



Obrázek 4.4: Poloha zařízení při pořizování snímku a příslušná EXIF informace [3]

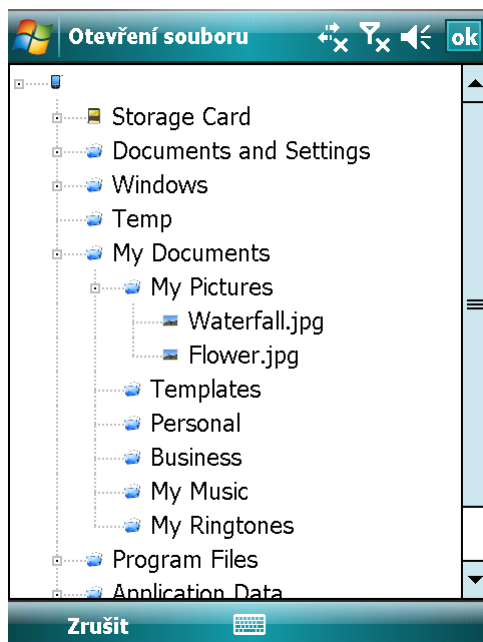


(a) Systémový dialog pro otevření obrázku s náhledy SelectPictureDialog

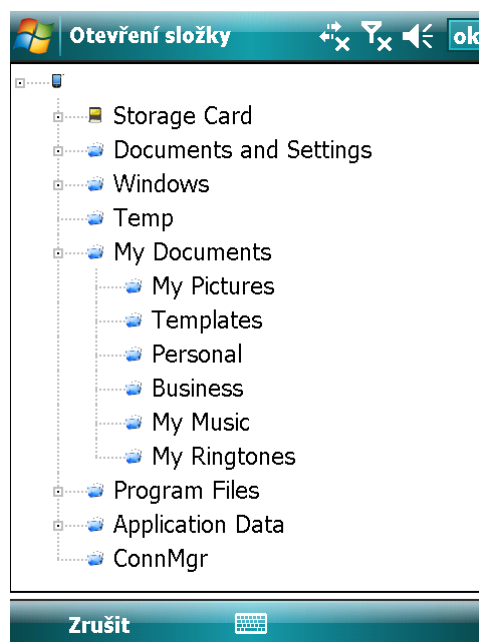


(b) Systémový dialog pro otevření souboru OpenFileDialog

Obrázek 4.5: Systémové dialogy pro otevření obrázku



(a) Dialog pro otevření obrázku



(b) Dialog pro otevření složky

Obrázek 4.6: Vlastní dialogy

## Kapitola 5

# Testování aplikace

Pro ověření funkčnosti aplikace je velmi důležité provést testování. Skládá se jednak z vyzkoušení prohlížeče na různých zařízeních a kontroly, zda všechny funkce vykonávají zadané úkoly. Z hlediska použitelnosti aplikace se potom zjišťují hlavně paměťové nároky, přesnost systému pro rozpoznávání gest a rychlost odezvy programu.

Kapitola se zabývá jednak použitými zařízeními pro testy, následuje popis navržených testů a jejich vyhodnocení.

### 5.1 Použitá zařízení

Na otestování byly použity dva mobilní telefony s operačním systémem WM 6.1. Jedná se o HTC Hermes (obr. č. 5.1(a)) a novější HTC Touch Diamond (obr. č. 5.1(b)). V tabulce č. 5.1 jsou vypsány základní parametry těchto telefonů. Liší se hlavně rozlišením displeje, taktem procesoru a velikostí operační paměti RAM.

HTC Touch Diamond obsahuje pohybový a polohový senzor, byl tedy využit k otestování rotace obrazovky podle polohy.

	HTC Hermes	HTC Touch Diamond
Frekvence procesoru	400MHz	528MHz
Velikost operační paměti	64MB	192MB
Volná operační paměť	35MB	45MB
Rozlišení fotoaparátu	2MPx	3MPx

Tabulka 5.1: Vybrané parametry použitých zařízení

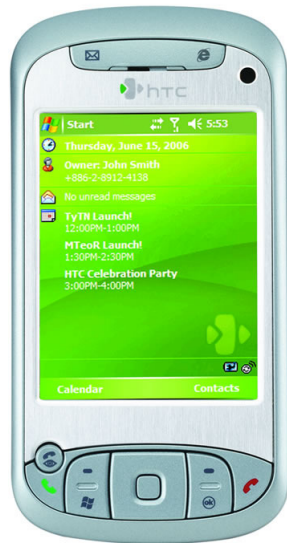
### 5.2 Navržené testy

Pro otestování podpory všech zadaných formátů byl použit obrázek ve formátu JPG pořízený digitálním fotoaparátem s rozlišením 4MPx. V počítači byl obrázek převeden do formátů BMP, PNG a GIF.

Pro otestování funkčnosti použití alba bylo zvoleno otevření a načtení složky se 100 obrázky různé velikosti i formátu.

Pro vyzkoušení funkčnosti otáčení obrázku podle EXIF informace byl vytvořen speciální obrázek s tímto příznakem.





(a) HTC Hermes



(b) HTC Touch Diamond

Obrázek 5.1: Mobilní zařízení použita pro testování aplikace

Program by měl bez problému zvládat jakékoli změny v nastavení. Při úpravě nastavení by se tedy změny měly projevit okamžitě, ale také zůstat zachovány při opětovném spuštění aplikace.

Nebezpečnou chybou při tvoření aplikace je jistě nekorektní práce s pamětí. *Garbage collector* se sice postará o nevyužitou paměť, ale často až po určité době. Mezitím ale program může požadovat další část paměti a dojde k výjimce *OutOfMemoryException*. V aplikaci je toto ošetřeno ručním uvolňováním větších částí paměti hned, jakmile nejsou potřeba. Například při otočení telefonu dochází ke změně orientace, starý obrázek se musí uvolnit a namísto něj se načte nový s aktuálním rozlišením. Je tedy vhodné vyzkoušet rychlý sled úkonů načítání obrázků a otáčení telefonu, které by měly odhalit případné problémy.

Paměťové nároky, zvláště v mobilních zařízeních, jsou jedním z důležitých kritérií při výběru prohlížeče. Je tedy vhodné zjistit využití paměti jednak při automatickém zarovnání a také při maximálním přiblížení obrázku.

Velmi důležitým kritériem pro výběr prohlížeče je rychlost jeho odezvy. Proto je vhodné otestovat různé úkony a změřit čas jejich provedení.

Při nedostatku volné paměti prohlížeč nemůže korektně zobrazovat obrázky. Mělo by ale vypsat chybové hlášení a nezpůsobit pád aplikace.

Rozpoznávání gest by mělo vyhovovat pokud možno všem uživatelům. Prohlížeč byl tedy vyzkoušen několika osobami, které provedly desetkrát tři různá gesta - šikmá čára, čtverec a vodorovná čára. Na konci se vyhodnotil počet úspěšných rozpoznání.

### 5.3 Vyhodnocení testů

Při testování funkčnosti načítání čtyř zadaných formátů jsem narazil na chybu. Knihovna *Drawing.dll* ze SDF obsahuje nástroj pro získání základních informací. Zjišťuje také počet

bitů na pixel. Při načítání testového obrázku ve formátu gif ale vrátila nesmyslnou hodnotu. Přidal jsem tedy kontrolu na tuto chybu a dodatečné nastavení na 24 bitů na pixel.

Otestování funkčnosti alba, stejně tak otáčení obrázku podle EXIF informace proběhlo úspěšně.

Při testování nastavení bylo změněno zobrazení hlavní obrazovky, byly zvoleny jiné nástroje, zakázána rotace displeje podle polohy zařízení, změněno nastavení prezentace, definovány nové tvary gest a byl zvolen jiný typ dialogu pro otevírání souborů. Byl tak pokryt celý rozsah nastavení. Program změny přijal a zpracoval korektně.

Při testování odolnosti programu na velké zatížení nebyly zaznamenány žádné problémy. Přestože načítání velkých obrázků s plným přiblížením trvá poměrně dlouho (několik sekund), program vždy provedl to, co bylo zadáno.

Pro zjištění paměťových nároků bylo využito nástroje *Správce úloh* a odečteny přesné hodnoty využití paměti prohlížečem.

	HTC Hermes	HTC Diamond
Automatické zarovnání (MB)	2.11	1.36
Maximální přiblížení (MB)	7.71	6.98

Tabulka 5.2: Využití paměti při načteném obrázku

Z tabulky č. 5.3 můžeme pozorovat rozdíl v odečtených hodnotách mezi telefony. Pravděpodobně se jedná o jiný způsob měření. Při automatickém zarovnání tak program zabírá 1.36 MB (resp. 2.11 MB). Při maximální přiblížení byly hodnoty 6.98 MB (resp. 7.71 MB). V obou případech jsou hodnoty odpovídající a přiměřené. Z hlediska paměťové náročnosti tedy implementovaný program vyhovuje.

	HTC Hermes	HTC Diamond
Načtení obrázku z vlastní kamery se zarovnáním (s)	1.2	1.6
Načtení obrázku z vlastní kamery bez zarovnání (s)	2.6	3.0
Načtení obrázku 0.3 MPx se zarovnáním (s)	0.6	0.5
Načtení 4 MPx obrázku bez zarovnání (s)	6.3	3.6
Načtení 4 MPx obrázku se zarovnáním (s)	2.2	1.9
Rotace 4 MPx obrázku bez zarovnání o 90° (s)	6.3	5.9
Rotace 4 MPx obrázku se zarovnáním o 90° (s)	neměřitelné	neměřitelné

Tabulka 5.3: Naměřené rychlosti odezvy programu

Pro testování rychlosti odezvy byly vybrány vzorové úkoly, u kterých se předpokládá časté využití a také úkoly se značnou výpočetní náročností. Naměřené výsledky jsou vypsány v tabulce č. 5.3. Načítání obrázků z vlastního fotoaparátu telefonu se podle druhu zarovnání pohybovalo mezi 1.2s a 3.0s. Načtení obrázku s malým rozlišením 0.3MPx proběhlo přibližně za 0.5s. Při načítání většího 4MPx obrázku se projevil rozdíl ve výkonnosti mezi telefony, zatímco HTC Hermes načel obrázek bez zarovnání za 6.3s, HTC Diamond jej načel za 3.6s. Se zarovnáním byly hodnoty podstatně nižší a to 1.9s (resp. 2.2s). Naměřené časy jsou poměrně příznivé a v tomto směru je prohlížeč naprosto použitelný. Při rotaci obrázku velmi záleží na aktuálním přiblížení. Při maximálním přiblížení trvala rotace o 90° 5.9s (resp. 6.3s). Rotace obrázku se zarovnáním byla téměř okamžitá. I v tomto ohledu je prohlížeč bez problému použitelný.

Pro zjištění chování programu při nedostatku paměti byl telefon zatížen různými apli-

kacemi. Při zkoušení načítání obrázku bylo zjištěno, že WM ukončuje neaktivní procesy, pokud volná paměť klesne pod 10MB. Prohlížeč fungoval korektně a správně načel celý obrázek.

Pro otestování rozpoznávání gest byl telefon s programem předán pěti lidem, kteří zkoušeli různá gesta. Úspěšnost rozpoznání se pohybovala kolem 90%. Systém rozpoznání gest je tedy vyhovující.

Z těchto úspěšně zvládnutých dílčích testů s lze vyvodit závěr, že aplikace je prakticky upotřebitelná v běžném použití.

# Kapitola 6

## Závěr

Závěrečná kapitola se věnuje především srovnání s již existujícími prohlížeči, dále je zde rozebráno, co nového práce přinesla. Věnuje se navrhovaným vylepšením a budoucnosti programu.

### 6.1 Porovnání s existujícími prohlížeči

Existující prohlížeče obrázků byly představeny v kapitole 3. Vytvořený prohlížeč obrázků vyniká především v možnosti konfigurace ovládání a nastavení. Nabízí možnost vytvářet vlastní alba, jejichž náhledy lze zobrazit na hlavní obrazovce. Mezi obrázky lze potom lehce přepínat kliknutím na náhled. Jako jediný z programů umožňuje konfigurovat ovládací panel, uživatel si tedy může zvolit nastavení, které mu vyhovuje nejvíce. Největší odlišností a výhodou tohoto programu je použití gest pro ovládání a možnost jejich konfigurace.

Vestavěný prohlížeč obrázků ve WM disponuje příjemnou možností přiblížit i velký obrázek na maximální rozlišení. Tento prohlížeč je určen pro vykonávání základních úkonů, nenabízí žádné pokročilé funkce.

Vestavěný HTC prohlížeč plní obdobnou funkci jako vestavěný WM prohlížeč. Je navržen modernějším způsobem s příjemnějším designem. Jeho výhodou je integrovanost do grafického prostředí telefonů HTC. Také ale nepočítá s pokročilými funkcemi.

Picture Viewer je zajímavý koncept, který se snaží implementovat ovládání pomocí gest. Dle mého názoru je ale velmi nedotažený a není dobře použitelný.

Dava Picture Viewer je poměrně dobře použitelný prohlížeč s podporou tvorby alb. Nedisponuje sice moderním vzhledem, ale je velmi intuitivní.

Resco Photo Manager je velmi profesionální program podporující pokročilé funkce. Vyniká především svou rychlostí a snadností ovládání. Bohužel nepodporuje použití gest.

### 6.2 Přínos práce

Podstatou této bakalářské práce bylo vytvořit konfigurovatelný prohlížeč obrázků pro WM. Zadání práce jsem vybíral v roce 2010, tj. v době, kdy ještě systém Windows Phone 7 nebyl vůbec rozšířený a nebylo zřejmé, zda vývoj Windows Mobile bude pokračovat. Po prostudování existujících řešení a seznámení se s vývojem pro mobilní zařízení jsem navrhl uživatelské rozhraní aplikace. V průběhu práce jsem narazil na různé problémy, jako je nedostatek paměti, neexistence dialogu pro otevření složky či neschopnost systému pracovat s gesty. Všechny tyto problémy se mi podařilo úspěšně vyřešit a získal jsem cenné zkušenosti.

Vypracovaný prohlížeč doplnil řady existujících programů. V moderních systémech je ovládání gesty naprostou samozřejmostí, v tomto ohledu tedy prohlížeč přinesl moderní prvky ovládání i pro starší systém. Uživatel může dále využít možnosti vytvářet vlastní alba, pokud mu tento způsob nevyhovuje, nemusí je vůbec používat. Nastavení uživatelského rozhraní není možné u žádného ze zmiňovaných programů. Uživatel sám si může zvolit nastavení, které mu vyhovuje nejlépe.

### **6.3 Navrhovaná vylepšení**

Sociální sítě se stávají čím dál více populární. Zajímavé vylepšení by tedy byla možnost nahrávat fotografie z alb přímo na Facebook. Uživatelé by jistě ocenili i možnost poslání obrázku přes MMS. Dalším vylepšením by mohla být možnost jednoduché editace obrázků či možnost pořízení fotografie vestavěným fotoaparátem společně s GPS informacemi. Velmi neotřelým způsobem ovládání by mohlo být využití pohybového senzoru pro gesta.

### **6.4 Budoucnost programu**

Program bude zveřejněn na internetu na stránkách *giw.webnode.cz* pod licencí *freeware*. Pokud bude ze strany uživatelů o prohlížeč zájem, bude přeložen i do anglického jazyka a bude nadále vyvíjen.

# Literatura

- [1] Accusoft Corporation: ImageGear for .NET Compact Framework.  
<http://www.accusoft.com/ig-compactframework.htm>, 2012 [cit. 2012-1-10].
- [2] Accusoft Corporation: Runtime & Deployment Licensing.  
<http://www.accusoft.com/licensing.htm>, 2012 [cit. 2012-1-10].
- [3] Impulse Adventure: JPEG Rotation and EXIF Orientation.  
<http://www.impulseadventure.com/photo/exif-orientation.html>, 2008 [cit. 2012-5-3].
- [4] Koushik Dutta: Using HTC Diamond.s Accelerometer/Sensor SDK from Managed Code. <http://blog.enterprisemobile.com/2008/07/using-htc-diamonds-sensor-sdk-from-managed-code/>, 2008 [cit. 2012-5-3].
- [5] Koza, P.: Windows Phone 7 Series - shrnutí, zamyšlení [online].  
<http://www.ce4you.cz/articles/detail.asp?p=0&a=361&pc=>, 2010-02-18 [cit. 2012-5-3].
- [6] Lead Technologies: LEADTOOLS Licensing SDK.  
<http://www.leadtools.com/corporate/licensing.htm>, 2012 [cit. 2012-1-10].
- [7] Lead Technologies: LEADTOOLS Mobile Imaging Developer Toolkit.  
<http://www.leadtools.com/sdk/mobile-imaging.htm>, 2012 [cit. 2012-1-10].
- [8] Microsoft: Windows Mobile 6 Professional and Standard Software Development Kits Refresh. <http://www.microsoft.com/en-us/download/details.aspx?id=6135>, 2007 [cit. 2012-5-11].
- [9] Mikudík, R.: Z pravěku do přítomnosti: historie systému Windows Mobile [online].  
[http://mobil.idnes.cz/z-praveku-do-pritomnosti-historie-systemu-windows-mobile-pn9-/mob\\_tech.aspx?c=A071126\\_121747\\_tech-a-trendy\\_ram](http://mobil.idnes.cz/z-praveku-do-pritomnosti-historie-systemu-windows-mobile-pn9-/mob_tech.aspx?c=A071126_121747_tech-a-trendy_ram), 2007-12-14 [cit. 2012-5-3].
- [10] Mikudík, R.: Historie systému Windows Mobile, díl druhý [online].  
[http://mobil.idnes.cz/historie-systemu-windows-mobile-dil-druhy-f2f-/aplikace.aspx?c=A071213\\_235330\\_programy\\_ram](http://mobil.idnes.cz/historie-systemu-windows-mobile-dil-druhy-f2f-/aplikace.aspx?c=A071213_235330_programy_ram), 2007-12-19 [cit. 2012-5-3].
- [11] Nikola Čech: Grafické formáty I - JPG a PNG.  
<http://www.emag.cz/graficke-formaty-i-jpeg-a-png/>, 2007-01-25 [cit. 2012-5-3].

- [12] Nikola Čech: Grafické formáty II - BMP, GIF, RAW a ostatní. <http://www.emag.cz/graficke-formaty-ii-bmp-gif-raw-a-ostatni/>, 2007-01-31 [cit. 2012-5-3].
- [13] OpenNETCF Consulting: OpenNETCF Shared Source License. <http://www.opennetcf.com/sharedsourcelicense.ocf>, 2012 [cit. 2012-5-3].
- [14] OpenNETCF Consulting: Smart Device Framework. <http://www.opennetcf.com/Products/SmartDeviceFramework/tabid/65/Default.aspx>, 2012 [cit. 2012-5-3].
- [15] Simon Mc Kenzie: ExifLib – A Fast Exif Data Extractor for .NET 2.0+. <http://www.codeproject.com/Articles/36342/ExifLib-A-Fast-Exif-Data-Extractor-for-NET-2-0>, 2012 [cit. 2012-5-3].
- [16] Wigley, A.: *Microsoft Mobile Development Handbook*. Microsoft Press, 2007, iSBN 0735623589.
- [17] WWW stránky: Open Icon Library. <http://openiconlibrary.sourceforge.net/>, 2010 [cit. 2012-5-11].

# Příloha A

## Tok obrazovek

