

Editor fotografií

Projekt ITU, 2015Z

Číslo projektu: 13

Číslo a název týmu: 117. Tým xpitri00

Autor: Jakub Pastuszek (xpastu00),

Další členové týmu: Martin Pitřík (xpitri00), Vojtěch Průša (xprusa05)

Termín řešení: 21. 9. - 18. 12. 2015

Abstrakt

Cílem projektu je navrhnout editor fotografií. Hlavním cílem je uživatelsky intuitivní, nikterak rušivé a rychle použitelné grafické uživatelské rozhraní (dále GUI). Důraz je kladen hlavně na přehlednost a jednoduchost výsledné aplikace. Dále je důležité informovat uživatele o operacích, které aplikace provádí a dát uživateli možnost vrátit akce zpět.

Zadání

Navrhněte a implementujte jednoduchý editor fotografií. Zaměřte se na intuitivnost a efektivnost přístupu k různým funkcím programu - aplikace filtrů, apod. Pro realizaci funkčních prvků aplikace využijte existující knihovny (např. OpenCV).

Cílové požadavky na aplikaci a její rozhraní

- Cílem aplikace je vytvoření GUI s Back-endem pro snadnou úpravu fotografií.
- Splnění cílů bude zjišťováno pomocí testování uživatelů při práci s aplikací, případně podle jejich zpětné vazby.
- Aplikace má primárně sloužit běžným uživatelům pro snadnou editaci fotografií, ale budou ji moci využít i zkušenější pro její rychlost úprav.
- Uživatel a jeho seznámení se aplikací (s jejím uživatelským rozhraním) by se nemělo potýkat s problémy typu dlouhé čekání bez znázornění toho, že aplikace něco provádí - nesmí nastat pocit zatuhnutí aplikace.
- Uživatel by měl pomocí metafor snadno rozpoznat, kterou funkci chce použít [1].
- Spokojený uživatel je ten, který nalezne v krátké době funkce, které od programu požaduje a aplikaci používá opakovaně.

Studium uživatelů, UI a testování

- Běžný uživatel nezná využití klávesových zkratk, ale umí využívat klávesnici a myš, proto musí být veden GUI pomocí těchto příslušenství. Uživatel kliká na to co mu je povědomé, nebo co ho zajímá - např. otáčení obrázků, zobrazování histogramu - co se stane když..? Rozhraní může obsahovat také již zmíněné klávesové zkratky, pro ty zdatnější uživatele, což může urychlit/usnadnit ještě více práci s aplikací.
- Podobný problém řeší aplikace jako např. Zoner Photo Studio¹ nebo Adobe Photoshop². Tyto aplikace jsou ale mnohem komplexnější a proto pro běžného uživatele mohou být značně nepřehledné a vyhledávání určitých funkcí bývá dosti zdlouhavé.
- Testování může probíhat pomocí anket např. ohledně rozmístění aktivních prvků, možnosti implementace funkcí, variabilní nástrojová lišta aj. Dále výběrem vzhledu tmavé či světlé, ikony jednotné, barevné nebo monochromatické. Při beta testech se dá uživatele „sledovat“, buďto přímo vizuální kontrola (co dělá a jak to dělá), nebo zpětnou vazbou - které akce uživatel provedl a v jakém pořadí, které funkce použil, využil možnosti vrácení provedeného kroku a další. Dále možnost zhodnocení po testování zpětnou vazbou v podobě zprávy nebo opět nějakého dotazníku.
- Uživatel, který je spokojený s aplikací může projevovat zájem o účast na jejím dalším testování, může jí různě recenzovat, propagovat. Doporučení je určitě dobrý předpoklad pro možnost rozšíření mezi více uživatelů.

¹ <https://www.zoner.cz/>

² <https://www.adobe.com/cz/products/photoshop.html>

Návrh GUI

- Po analýze vystavených anket, kde odpovídali různí respondenti (viz příloha) se ukázalo, že uživatelé by chtěli spíše tmavší GUI (viz Graf 1), dále mezi nejžádanější funkce patřily transformace obrazu, a změna jasu a kontrastu (viz Graf 2). Další anketa směřovala na vzhled ovládacích prvků, kde jednoduché prvky převažovaly nad složitějšími (viz Graf 3). A v posledním řadě, rozhraní jednookenní, je přednější než rozděbené do více rozdělených oken (viz Graf 4).
- Uživatel otevírá a ukládá fotografie, používá různé filtry a funkce, přibližuje popř. oddaluje fotografie.
- Jednou z variant rozhraní může být fotografie přes celé okno s vysouvacími se ovládacími prvky, což zajišťuje co možná největší oblast pro zobrazení fotografie, uživatele neruší žádné elementy. Toto je vhodné např. pro zobrazení po aplikování filtru a zhodnocení výsledku, nebo se stále zobrazenými ovládacími prvky, kdy se fotografie trochu zmenší, ale interakce s ovládacími prvky bude rychlejší.



Obrázek 1

Návrh testování

- Mezi klíčové prvky GUI patří návrh ovládacích prvků, kde ty nejpoužívanější musí být co nejrychleji a nejjednodušeji dosažitelné a dále co možná největší plocha pro zobrazení fotografie.
- Efektivita zobrazení fotografie se dá vylepšovat upravením hodnoty přiblížení pro co největší vyplnění volné plochy. Dále rychlostí provádění filtrů, úprav a načítání samotné fotografie.
- Testování výsledné aplikace proběhne na stejném vzorku uživatelů, kteří přispívali do anket, jedná se o běžné uživatele.
- Do testování se budeme snažit pobídnout co nejvíce dalších uživatelů, abychom dostali co nejdůvěryhodnější výsledky.
- Testeři budou řešit úlohy od nejlehčích po složitější, tzn. postupně se s aplikací seznámí, tím pádem složitější úlohy budou lépe zvládat.

Studium realizace GUI

- Vhodné pro realizaci jsou jakékoli desktopové technologie, jako např. WPF, Qt aj. Naopak webové technologie jsou celkem nevhodné pro tuto aplikaci, ale ve zvláštních situacích mohou být použity také.
- Technologie WPF s jazykem C# v pozadí je vhodná zejména kvůli datové struktuře „bitmapy“, která ulehčí uchování fotografie v paměti a také různé funkce pracují s touto strukturou, proto práci usnadňují. Z hlediska GUI bitmapové struktury se dají snadno zobrazit. Tato technologie je použitelná jen na OS Windows. Zatímco technologie Qt je multiplatformní. Qt technologie nenabízí struktury „bitmapy“, proto pro nás není tak zajímavá, i když také nabízí spousty usnadňujících prvků, WPF je pro naši aplikaci vhodnější. Webové technologie nám neposkytují téměř žádná usnadnění, proto se touto cestou realizace nevydáme.

Návrh/popis back-endu

- Dostatečná funkčnost pro testování GUI aplikace bude splněna po implementování načítání fotografie do paměti a tudíž i vykreslení histogramu, který je součástí GUI. Další ikony funkcí pro editaci budou viditelné, budou mít titulky, ale nebudou funkční, ale to z hlediska testování GUI není problém.
- Pro správnou funkci je třeba vykonat funkci pro načtení fotografie do paměti aplikace, aby se zobrazila a také aby se načel histogram a daly se použít různé filtry a aby bylo možné samotnou fotografii jakkoli upravovat.
- Při každém použité filtru na fotografii je potřeba přepočítat histogram, aby odpovídal výsledku. Dále je hodně důležitá historie úprav, kdy se každá provedená akce zobrazí a v případě nespokojenosti může být vzata zpět. Tato historie akcí je omezena na posledních 10 změn. Toto dodává uživateli určitou důvěru, že když něco „pokazí“, tak se tato akce dá vrátit.
- Všechny zobrazené/použitelné funkce (jejich piktogramy nebo textová zastoupení) budou plně funkční, funkce budou vracet smysluplné výsledky.
- Back-end je na GUI napojen tak, že akce se vyvolává přes GUI „klikátka“ (piktogramy, ikony, textové symboly aj.), je zpracována back-end funkcemi a zpátky se změny projeví v GUI (např. Překreslení fotografie podle filtru, překreslení histogramu).

Programování back-endu

Třída `Filters` obsahuje jednotlivé filtry a další metody pro editaci bitmapy. Třída obsahuje parametrický konstruktor, který přijímá referenci na zpracovávanou bitmapu. Dále následují tyto metody:

- `void setContrast(int threshold)` metoda provede nastavení kontrastu dle zadané prahové hodnoty v intervalu od -100 do 100
- `void setBrightness(int brightness)` metoda provede nastavení jasu dle zadaného parametru v intervalu od -255 do 255
- `void toGrayScale()` metoda provede konverzi barevné bitmapy na bitmapu v odstínech šedé
- `void toInvert()` metoda provede inverzi barev bitmapy
- `void toBlackAndWhite(int threshold)` metoda provede konverzi barevné bitmapy na bitmapu v černobílých barvách dle zadaného prahu v intervalu od 0 do 255
- `setNewSize(int width, int height)` metoda pro změnu velikosti bitmapy
- `rotate(rotation)` metoda pro rotaci bitmapy
- `flip(flip)` metoda pro překlopení bitmapy
- `blur(int effect)` metoda pro rozmazání obrazu dle zadané síly efektu v intervalu od 0 do 255
- `toOldPhotograph()` metoda pro převedení bitmapy do bitmapy s efektem staré fotografie
- `addColoredNoise()` metoda pro přidání barevného šumu do bitmapy

Třída `Histogram` slouží k analýze bitmapy a k zobrazení jejího histogramu. Třída obsahuje parametrický konstruktor, který přijímá referenci na zpracovávanou bitmapu. Dále následují tyto metody:

- `void createHistogram()` metoda vypočte histogram
- `Bitmap drawHistogram()` metoda vykreslí histogram do bitmapy, kterou posléze vrací jako svou návratovou hodnotu

Třída `Navigation` slouží pro vytvoření náhledu a navigaci po editované bitmapě. Třída obsahuje parametrický konstruktor, který přijímá zpracovávanou bitmapu. Dále jsou ve třídě obsaženy následující metody:

- `Bitmap drawNavigation()` metoda vykreslí navigaci v bitmapě

Programování GUI

- Pro tvorbu GUI bylo využito návrháře Windows Forms ve Visual Studiu 13.
- Prvky GUI jsou ve většině případů řešeny pomocí tlačítek, kdy každé z nich vyvolá určitou událost, která se provede nad upravovaným obrázkem a také se provede její zapsání do historie událostí. Další možnosti spuštění událostí (hlavně filtrů) je v horní nabídce. Zobrazení mezi histogramem a navigátorem je řešeno pomocí tabů (oušek), kdy jedno zobrazení překrývá druhé a naopak. Pro informování uživatele o stavu provádění dané akce je v aplikaci zobrazen indikátor průběhu (ang. progress bar).
- Jelikož pro vývoj aplikace byl vybrát již zmíněný Windows Forms, tak oddělení back-endu od front-endu(GUI) je značně složité, proto se kod pro GUI nachází hned za kódem pro back-end. V důsledku tohoto se z hlediska jednoduchosti úprav ubírá na přehlednosti, nelze aby na stejném modulu pracoval koder i programátor.
- Mezi nejpoužívanější bloky patří popisy (Label), tlačítka (Button), rámce (GroupBox) – pro správné rozložení prvků aj.

Testování

- Testování probíhalo na skupině 5ti uživatelů z cílové skupiny. Testování bylo velmi přímočaré, kdy každý testovaný uživatel dostal stejných 5 úkolů, které měl vypracovat po krátkém seznámení se s aplikací. Uživatelské pohyby myši a stisky kláves byly zaznamenávány a následně společně vyhodnoceny (viz Výsledky). Po dokončení všech úkolů následoval dotazník, kde měl každý zúčastněný odpovědět na pár otázek ohledně přehlednosti a užitečnosti aplikace (viz Přílohy-Testování). Dále byly v dotazníku obsaženy otázky ohledně složitosti jednotlivých úkolů.
- Také byla aplikace rozeslána dalším 8mi uživatelům, kteří měli za úkol se s aplikací seznámit a otestovat ji dle svých preferencí na funkcionálnost a poté ji stejně jako pětice důkladněji testovacích uživatelů ohodnotit pomocí vyplnění dotazníku na přehlednost a užitečnost (viz Přílohy-Testování).

Výsledky a závěr

- Výsledky testování vynesené do grafů lze nalézt v Příloze B.
- Testování uživatelé byli ve více než dvou třetinách spokojeni se vzhledem aplikace. S dotazem jak působí přehlednost rozhraní aplikace naprostá většina dotazovaných odpověděla pozitivně. Dle nabídky funkcí se již názory více rozcházejí. Více než třetina uživatelů očekává více funkcí od výsledné aplikace. Samotný průzkum aplikace a její „porozumění“, s tím měla celá jedna třetina dotazovaných problém. S ohledem na poslední dotaz, je potřeba aplikaci více „přiblížit“ uživatelům, dotaz nějaké tipy na používání a také nápovědu.
- Výsledky testů aplikace splňují očekávání. Mezi vylepšení aplikace by se dala zařadit lokalizaci UI, či změna ikon.

Týmová spolupráce

- Týmová spolupráce byla přínosem hlavně v oblasti návrhu, kdy každý z týmu nastudoval potřebnou literaturu a prošel podobná řešení a své výsledky prezentoval ostatním, kde se následnou diskuzí rozhodlo o nejlepší variantě se kterou se pokračovalo dále.
- Jako užitečné bylo rozdělení implementace do dvou samostatných oddílů (tj. backend a frontend), a ty dále dělit na jednotlivé moduly. Rozdělení mezi členy týmu bylo poté jednoduché a to tak, že každý pracoval na svém modulu (jeden větší nebo více malých) a poté spojením všech vznikla výsledná aplikace. Úskalím tohoto řešení bylo to, že jsme museli navrhnout společné rozhraní, aby moduly spolu mohly správně komunikovat.
- Mezi další výhodou by se dalo určitě zařadit možnost věnovat se tomu co každému nejvíce vyhovuje. Jeden měl lepší nápady na GUI, druhý zase lépe zvládal programování backendu. V řešení projektu jsme se vzájemně doplňovali.
- Jako nevýhoda by se dal brát fakt, že se občas musí čekat na vyhotovení modulu jiným členem týmu.

Závěr

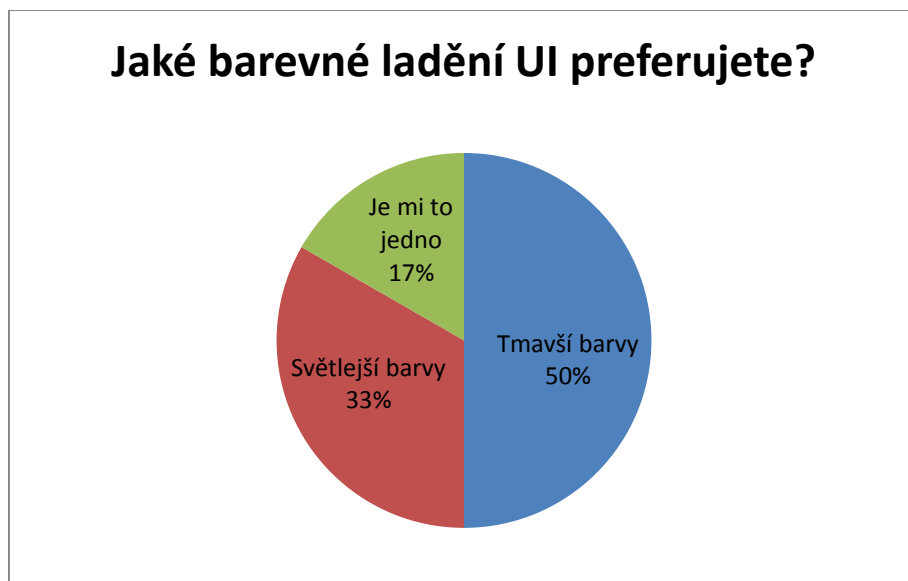
- Výsledná aplikace splňuje požadavek na intuitivní ovládání, který byl vyhodnocen z výsledků testování. Aplikace také splňuje přehlednost a rychlost proveditelnosti operací, kdy nejpoužívanější funkce jsou uživateli zobrazeny a nabídnuty rovnou bez dalších nadbytečných operací (klikání, otevírání nabídek aj.). Umožnění vrácení provedené akce je uživateli velmi vítaná funkcionalita.

ODKAZY

- [1] FADEYEV, Dmitry. *8 Characteristics Of Successful User Interfaces* [online]. April 15, 2009 [cit. 2015-10-18]. Dostupné z: <http://usabilitypost.com/2009/04/15/8-characteristics-of-successful-user-interfaces/>

PŘÍLOHY

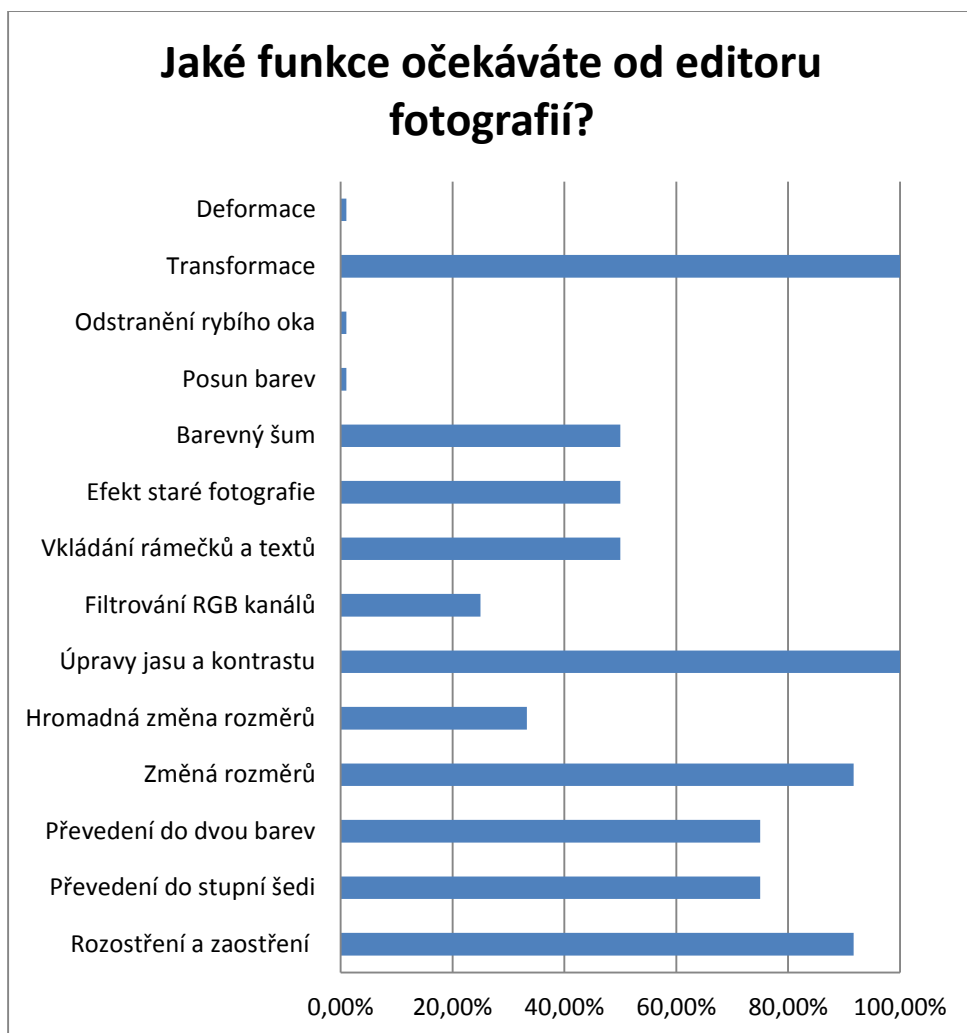
A - pre-test



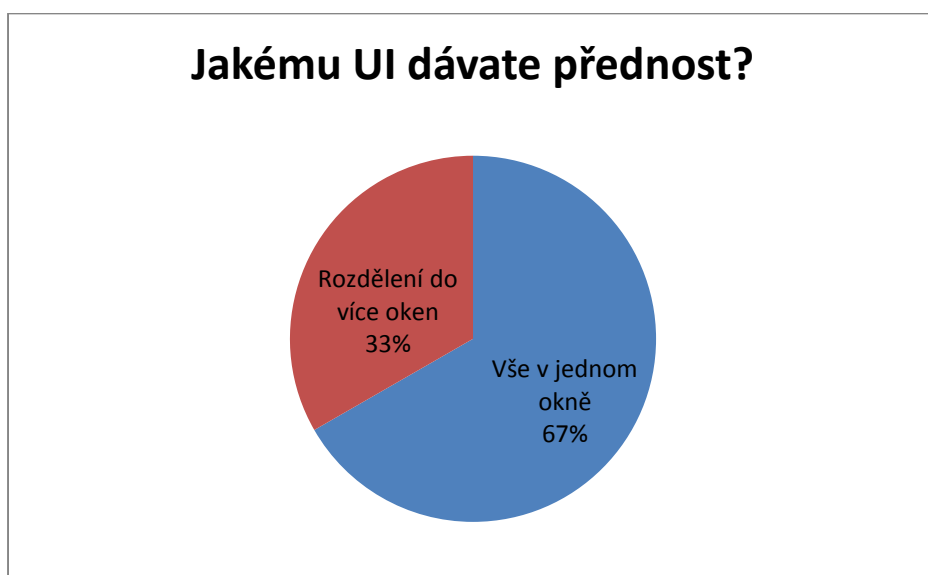
Graf A.1 - Barvy GUI



Graf A.2 - Ovládací prvky



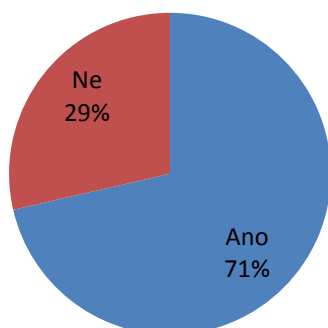
Graf A.3 – Funkce



Graf A.4 - Rozložení GUI

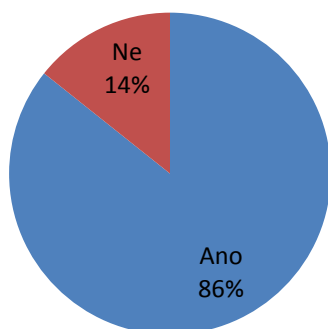
B - post-test

Splnil program očekávání ze strany vzhledu?



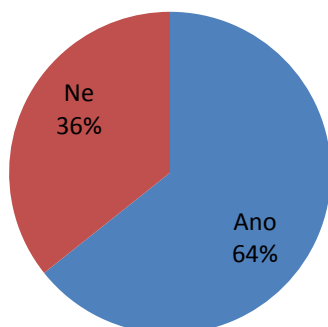
Graf B.1 – Očekávaný vzhled

Působí na vás rozhraní programu přehledně?



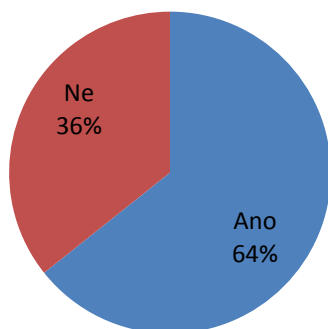
Graf B.2 – Přehlednost rozhraní

**Splňuje program očekávání ohledně
nabídky funkcí?**



Graf B.3 – Požadované funkce

**Bylo pro vás snadné se naučit z
programem pracovat?**



Graf B.4 – Intuitivnost ovládní