

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



**Univerzální nástroj pro rozšíření
možností redakční práce s obrazovým
materiálem**

Bakalářská práce

Filip Pýcha

Vedoucí práce: Novák Milan, Mgr. Ph.D.

České Budějovice 2013

Bibliografické údaje

Pýcha F., 2013: Univerzální nástroj pro rozšíření možností redakční práce s obrazovým materiálem [Universal tool for an extension of the editorial work with the pictorial materials] - p. 42, Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Bakalářská práce se zabývá vytvořením webové aplikace od samotné zadávací dokumentace přes vývoj a vysvětlení vnitřní struktury až po funkční aplikaci. Na úvod je čtenáři vysvětleno jaké základní pojmy, technologie a funkce budou použity ve výsledné aplikaci. Další část bakalářské práce se zabývá použitím vhodné metodiky vývoje webové aplikace, rozpracováním a popis zvolené metodiky. Finální část se věnuje návrhu, implementaci a následně prezentaci webové aplikace na redakčním systému VoxArea.

Klíčová slova

Grafický editor, redakční systém, nové technologie

Annotation

The bachelor's thesis deals with the creating of a web application from the beginning of the documentation over the developing and the explanation of the internal structure to the functional application. The introduction explains to the reader what the basic concepts, technologies and functions will be used in the resulting application. Next part of the bachelor's thesis deals with the use of the appropriate methodology for the development of the web application and the development and the description of the chosen methodology. The final section deals with the design, implementation and subsequent presentation of the web application CMS (Content Management System) VoxArea.

Key words

Graphics editor, content management system, new technologies

Prohlašuji, že svoji **bakalářskou** práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své **bakalářské** práce, a to **v nezkrácené podobě** elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 13.12.2013 Podpis autora

Obsah

1	Úvod	9
2	Cíle práce	9
3	Metodika	10
3.1	Prototypový přístup	11
4	Základní pojmy	12
4.1	CMS	12
4.2	Grafické editory	12
4.3	Grafický filtr	13
4.3.1	Zpracování obrazu	13
5	Zadání projektu	14
6	Analýza	15
6.1	Výběr funkcionalit a efektů	15
7	Použité technologie	18
7.1	Vývojové prostředí	18
7.2	HTML5	19
7.2.1	Canvas	20
7.3	CSS3	21
7.4	jQuery	22
8	Realizace grafického editoru	23
8.1	USE CASE diagram	23
8.2	CLASS diagram	23
8.3	Návrh designu	24
8.4	Programování designu	26
8.4.1	Menu	26
8.4.2	Problémy při vývoji aplikace	27
8.5	Programování funkcí	28

8.5.1	Načtení obrázku do Colorboxu	28
8.5.2	Přizpůsobení funkční části Colorboxu	30
8.5.3	Aplikování filtrů	30
8.5.4	Změna rozlišení	32
8.5.5	Ořez	32
8.5.6	Rotace	35
9	Testování	37
10	Závěr	38
A	Aplikace	42
B	Postup instalace	42

Seznam obrázků

1	Metodika - prototypový přístup	11
2	USE CASE Diagram	23
3	CLASS Diagram	23
4	Wireframe (drátěný model) aplikace	25

Seznam tabulek

1	Logický rámeček	14
2	Porovnání nástrojů a filtrů grafických editorů	16
3	Zavedení a vývoj funkcí ve FastPicture editoru	17
4	Příklad přístupu do canvasu	20
5	Import obrázku do canvasu	24
6	Zobrazování a skrývání menu	26
7	Varovná hláška při nekompatibilitě prohlížeče	27
8	Odstranění obal. elementů	28

9	Načítání obrázku do canvasu	30
10	První část: událost click	31
11	Druhá část: zjištění hodnot posuvníků a zavolání funkce na aplikování filtru	31
12	Třetí část: aplikování filtru	31
13	Zjištění hodnot a následné volání funkce na aplikování filtrů	32
14	Výpočet hodnoty cropStartX	33
15	Výpočet hodnoty cropStartY	33
16	Výpočet hodnoty cropEndX	34
17	Výpočet hodnoty cropEndY	34
18	Výpočet hodnoty cropWidth	34
19	Výpočet hodnoty cropHeight	34
20	Vytvoření dočasného canvasu a získání jeho contextu	35
21	Nastavení šířky a výšky dočasnému canvasu	35
22	Nastavení šířky a výšky primárního canvasu a finální vykreslení	35
23	Rotace canvasu ve směru hodinových ručiček	36
24	Rotace canvasu proti směru hodinových ručiček	36

1 Úvod

V dnešní době redakčních systémů je možnost publikovat obrázky k článkům či vytvářet webové galerie. Rychlé korektury obrázků se provádějí v offline editorech, což ale ubírá čas publicistům. Ti tak zároveň při korekturách mohou ztrácet hlavní myšlenku. Předpokládá se, že vytvořením projektu dojde k ulehčení a k zefektivnění publikace obrázků v redakčním systému, aniž by se používalo externích programů (Adobe Photoshop, Gimp, Paint.NET apod.) a takto se získal čas věnovat se jiným důležitějším částem.

V teoretické části této práce se čtenář seznámí se základními pojmy, které budou dále použity. Zmíněny jsou technologie, které jsou využity pro vývoj aplikace.

V dalších kapitolách práce se pojednává o jednotlivých funkcích a dalších možnostech rozšíření aplikace. V závěru teoretické části jsou popsány výhody a nevýhody vybraných technologií sloužících k vývoji této aplikace.

Praktická část se zaměřuje na vývoj aplikace od návrhu vývojových diagramů, samotného návrhu grafického designu až po vývoj rozhraní aplikace s jednotlivými funkcemi a filtry.

2 Cíle práce

Základním cílem bakalářské práce je vytvořit modul pro redakční systém, který umožní uživateli provést některé základní operace s obrázkem přímo před jeho odesláním na server v prostředí webového prohlížeče. Bude se jednat o univerzální nástroj, který umožní tento modul implementovat do jakéhokoli redakčního systému podporující rodinu HTML5 (HTML5, CSS3, JavaScript).

Díličí cíl pro splnění hlavní části zadání je analýza, díky které se zjistí, jaké offline grafické editory jsou nejpoužívanější. Na základě výběru grafických editorů je provedena analýza jednotlivých nástrojů a filtrů, které jsou použity u univerzálního grafického editoru.

Webová aplikace je poskytována uživatelům z webového serveru přes počítačovou síť, kde uživatel má možnost bez nutnosti instalace na lokální počítač spustit aplikaci přes webový prohlížeč, který nazýváme tenkým klientem. Výhody spočívají v rychlé opravě či vylepšení aplikace bez nutnosti aktualizace u všech uživatelů.

Možnosti výběru prostředí jsou velké, ale z hlediska multiplatformosti je zvoleno webové prostředí. Důvodem je podpora technologií ve všech zařízeních.

Celý tento projekt je vydán pod licencí GNU Affero General Public License [11], jenž umožňuje aplikaci bezplatně šířit, dále rozšiřovat, měnit a vytvářet různé plagiáty - dává možnost dalším programátorům převzít toto dílo a libovolně ho upravovat.

Ve finální fázi projektu vznikne softwarový produkt, který je nazván FastPicture editor na základě iniciál autora tohoto díla.

3 Metodika

Úvod praktické části je soustředěn na vývoj aplikace. V další fázi je použit testovací cyklus prototypový přístup.

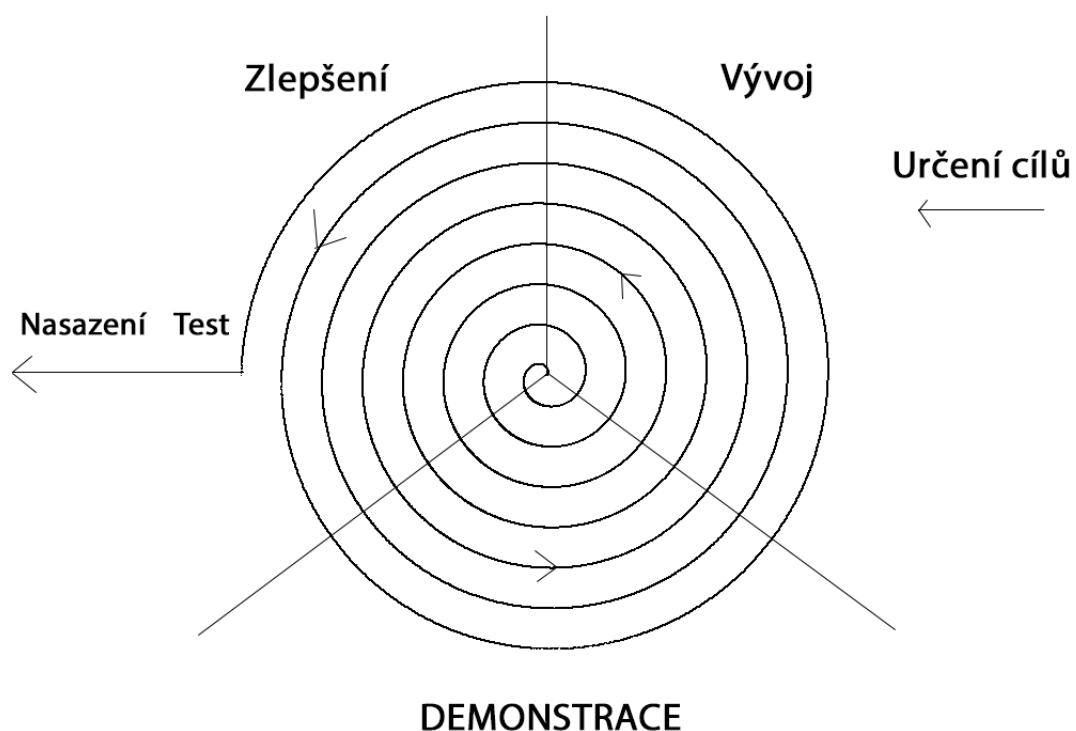
Pro úspěšný postup vývoje je potřeba vyhledat vývojový software (nejlépe Netbeans, Eclipse nebo jednoduchý textový editor) a libovolný operační systém (nezávislé na platformě).

V dalším kroku je nutné použít komparaci pro získání výsledků o nejpoužívanějších filtrech a nástrojích, stanovit primárního offline prohlížeče, poté porovnat s dalšími offline editory. Na základě získaných výsledků jsou vybrány vhodné filtry a nástroje.

Stanovení nejpoužívanějších filtrů je provedeno analýzou viditelnosti daných filtrů v nástrojových lištách a menu u grafických editorů. Pokud daný filtr se nachází na viditelném místě, tak je označen jako nejpoužívanější.

3.1 Prototypový přístup

Prototypový přístup je snaha snížit nebezpečí projektových rizik rozdělením projektu na menší části a zjednodušit tak možnost změn v průběhu procesu vývoje. Uživatel je zapojen v celém procesu vývoje, což zvyšuje pravděpodobnost přijetí konečné implementace uživatelem. Malé ukázky systému jsou vyvíjeny iterativním procesem, dokud se prototyp nevyvine tak, že splňuje požadavky uživatele. Většina prototypů je sice vyvíjena s tím, že budou vyřazeny, ale v některých případech je možné pokročit od prototypu k funkčnímu systému.



Obrázek 1: Metodika - prototypový přístup

4 Základní pojmy

4.1 CMS

Systém pro správu obsahu (CMS), neboli také často používané označení publikační systém či redakční systém, je určitý software, který umožňuje spravovat teoreticky libovolný obsah v závislosti na možnostech daného řešení. CMS je nejčastěji využíván pro správu webového obsahu, ovšem nejedná se o nutnou podmínku. Prostředí bývá rozdělené na uživatelskou (Front-End) a administrátorskou (Back-End) část.

Uživatelská část je určena široké veřejnosti a slouží k zobrazení obsahu. Administrátorská část je určena k veškeré obsluze obsahu i systému a měla by být dostupná pouze přihlášeným uživatelům s patřičným oprávněním. Mezi základní administrátorské funkce CMS patří správa jednotlivých sekcí systému, např. správa obsahu, uživatelů, nastavení atd. a v jednotlivých sekcích patřičné operace, např. přidání nových záznamů a editace, či smazání stávajících. Moderní CMS by měl být modulární, to znamená, že jednotlivé části systému by měli být samotné části, tzv. moduly, což umožňuje maximálně univerzální nasazení systému [28].

Všechny tyto systémy mají společnou vlastnost zveřejňovat textový a obrazový materiál s úpravou mimo systémy v offline grafických editorech. Tento grafický editor slouží jako modul, který má možnost integrace do těchto systémů pro rychlou korekturu obrazového materiálu.

4.2 Grafické editory

Grafické editory slouží k tvorbě a editaci obrazového materiálu. Grafické editory se dělí na dvě části a to podle toho s jakým typem obrazu pracuje.

Prvním typem obrazového materiálu je bitmapa, kdy obraz je složen z jednoduchých čtverečků tzv. pixelů vlastníci nějaký odstín barvy z různých barevných modelů (RGB, CMYK, HDR, atd.) [1].

Druhým typem obrazového materiálu jsou křivky tzv. vektory. Informace o vek-

torech jsou uchovány v textové podobě ve formě matematických vzorců.

4.3 Grafický filtr

Obecně bitmapové filtry fungují tak, že se postupně (po x a y souřadnici) procházejí všechny body matice obrázku a na tyto body se aplikuje vzorec filtru. Tento vzorec pak výstupnímu obrázku na souřadnicích x a y z původního obrázku přiřazuje novou barvu [29].

4.3.1 Zpracování obrazu

Rastrový obraz lze zpracovávat mnoha různými metodami s ohledem na požadovaný výsledný efekt.

Techniky, pomocí kterých jsou v počítači upravovány fotografie se nazývají zpracování obrazu (image processing). Při zpracování obrazu jsou existující obrazy nejprve převedeny do číslíkové podoby, a potom se zpětně zobrazují v grafické formě na displeji. Po převedení obrazu do číslíkové podoby je možno data dále zpracovávat.

Jako například vyhledávání informací v obraze, barevně odlišit segmenty obrazu na základě vstupních podmínek, změnit jas, ostrost, barevnost a mnoho dalšího. Zpracování obrazu je často používáno i v komerčních oblastech při retušování a skládání fotografií [30].

5 Zadání projektu

	Popis projektu	Objektivně ověřitelné ukazatele
Cíle projektu	Pro redakční systém vytvořit grafický editor s různými nástroji a efekty. S příjemným grafickým rozhraním a jednoduchou obsluhou.	Snížení časových nároků na úpravu obrázků, přehlednost, komfort.
Účel projektu	Úprava obrázků v prohlížeči bez nutnosti užití offline grafických editorů.	Snížení využití offline editorů.
Výstupy	Grafický editor (modul) pro redakční systémy, dokumentace.	Integrace editoru - prostudování dokumentace.
	Prostředky ověření	Předpoklady
Cíle projektu	Průzkum časových nároků před a po nasazení editoru na redakční systém.	Spokojenost administrátorů, přehlednost nástrojů.
Účel projektu	Elektronický dotazník spokojenosti administrátorů redakčních systémů.	Zjednodušení úprav, menší náklady na nákup editorů.
Výstupy	Výsledky funkčnosti na redakčním systému.	Zajištění správného nasazení.
Činnosti	Analýza offline grafických editorů Návrh a realizace aplikace Integrace do redakčního systému Testování	

Tabulka 1: Logický rámeček

6 Analýza

Analýza je provedena komparací, kdy jsou vybrány pro editor ty nejpoužívanější filtry a nástroje pro rychlou korekturu obrázků. Tyto grafické editory jsou zvoleny na základě průzkumu webových portálů, které umožňují stažení veškerého dostupného software včetně potřebných offline grafických editorů. Na základě statistiky, které udávají počty stažených produktů jsou vybrány offline grafické editory: Faststone Image Viewer z produkce FastStone soft, Picasa od firmy Google inc., Irfan View od vývojáře jménem Irfan Skiljan a poslední XnView od firmy XnSoft.

XnView je zvolen jako vzor pro porovnání se zbylými třemi editory z hlediska obsazení filtrů a základních nástrojů. Pokud daný filtr či základní nástroj je obsažen alespoň ve třech grafických offline editorech, jsou zahrnuty do nejpoužívanějších a následně implementovány do nové aplikace FastPicture editor.

V současné době se na trhu online grafických editorů nachází hned několik. V září r. 2013 firma Google inc. zavedla do svého projektu Google+ online grafický editor, který využívá obdobné technologie [24]. Dalším grafickým online editorem je odlehčená verze Photoshopu [25] od firmy Adobe, která běží na technologii Flash. Pixlr online grafický editor [26] od firmy Autodesk je také postaven na Flash technologii, který nabízí spoustu filtrů a nástrojů. Při analýze online grafických editorů nebyl nalezen žádný řádný grafický editor postaven na technologii Canvas.

6.1 Výběr funkcionalit a efektů

Na základě analýzy jsou sjednoceny běžné filtry a nástroje, které obsahují výše zmíněné offline grafické editory. Navíc jsou přidány další filtry, aby obohatily možnost další manipulace s obrázkem.

	Faststone	IrfanView	G. Picasa	XnView	Zavedeno
Úpravy					
Rotace	ANO	ANO	ANO	ANO	ANO
Rozlišení	ANO	ANO	NE	ANO	ANO
Oříznout	ANO	NE	ANO	ANO	NE
Retuš	ANO	NE	ANO	NE	NE
Červené oči	ANO	ANO	ANO	ANO	NE
Barvy					
Jas	ANO	ANO	ANO	ANO	ANO
Kontrast	ANO	ANO	ANO	ANO	ANO
Gamma	ANO	ANO	NE	ANO	ANO
Rozmazání	ANO	ANO	ANO	ANO	ANO
Šum	ANO	ANO	NE	ANO	ANO
Stupnice šedi	ANO	ANO	ANO	NE	ANO
Sépie	ANO	ANO	ANO	ANO	ANO
Negativ	ANO	ANO	NE	ANO	ANO
Filtry					
Pixelizace	ANO	ANO	ANO	ANO	ANO
Vodoznak	ANO	ANO	NE	ANO	NE
Reliéf	ANO	ANO	NE	NE	ANO
Posterizace	NE	NE	ANO	ANO	ANO
Olejomalba	ANO	ANO	NE	ANO	ANO
Solarizace	NE	ANO	NE	ANO	ANO
Medián	NE	ANO	NE	ANO	ANO
Detekce hran	NE	ANO	NE	ANO	ANO

Tabulka 2: Porovnání nástrojů a filtrů grafických editorů

Přidané filtry	Vývoj	Zavedeno
Kruhové rozmazání	NE	ANO
Křížové rozmazání	NE	ANO
Čtvercové rozmazání	NE	ANO
Rozptyl	NE	ANO
Chvění	NE	ANO
Olejomalba	NE	ANO
RGB kanály	NE	ANO
Ostrost	NE	ANO
Přírůstek / Zkreslení	NE	ANO
Inverzní barvy	NE	ANO
Maximum	NE	ANO
Medián	NE	ANO
Minimum	NE	ANO
Struktura	NE	ANO
Zrnění	NE	ANO
Průhlednost	NE	ANO
Sevření, vír	NE	ANO
Saturace	NE	ANO
Kaleidoskop	NE	ANO
Zkreslení objektivu	NE	ANO
Pilovité vlny	NE	ANO
Sinusové vlny	NE	ANO
Trojúhelníkové vlny	NE	ANO
Kruh	NE	ANO
Vodní vlny	NE	ANO
Twister	NE	ANO

Tabulka 3: Zavedení a vývoj funkcí ve FastPicture editoru

7 Použité technologie

K realizaci jednotlivých technologií je zvolen skriptovací jazyk JavaScript a jeho další členy tzv. rodiny HTML5 (HTML5, CSS3). K vývoji webové aplikace také neodmyslitelně patří redakční systém k nasazení dané aplikace do provozu.

7.1 Vývojové prostředí

Základní struktura aplikace je tvořena značkovacím jazykem HTML5, kdy tento standart je přísně dodržen. HTML5 je vybrán proto, že jeho podpora je přítomna u všech moderních zařízení od desktopových počítačů po mobilní sféru. Grafický design je navržen pomocí profesionálního grafického editoru Photoshop CS6 od firmy Adobe, který umožní navrhnout speciální efekty, různé přechody a poté následnou grafiku nařezat na dílčí elementy.

Kaskádové styly ve verzi 3 umožní veškerou nařezanou grafiku použít a následně nastylovat jednotlivé elementy značkovacího jazyka HTML5.

Celá funkční část aplikace je naprogramována skriptovacím jazykem JavaScript a její knihovnou jQuery, která je nápomocna při vývoji. jQuery je vybráno z důvodu jednodušší syntaxe oproti čistému JavaScriptu a velmi dobré dokumentaci.

Výsledná aplikace je navržena převážně pro desktopové počítače a notebooky s různými rozlišeními podporující výše zmíněné technologie. Podpora všech rozlišení je zajištěna pomocí Media queries [12], které spadají do kategorie kaskádových stylů, které jsou označovány jako tzv. responzivní design [13]. CSS3 je v současné době moderní stylovací nástroj, kterým jednoduše přizpůsobíme svoji aplikaci dnešním trendům.

Grafický editor je zkonstruován na základě zobrazovacího nástroje colorbox, který umožňuje po kliknutí na danou miniaturu zvětšit obrázek a přenést ho do popředí s tmavým podkladem. Tato kombinace barev a oddělovacích prvků je jedna z nejčitelnějších grafických podob a dodává člověku srozumitelnosti nad celou aplikací. Důraz je kladen na osoby trpící daltonismem [19].

7.2 HTML5

HTML5 je novým standardem představeným World Wide Web Consortiumem (W3C) a jeho pracovními skupinami. Je novou generací každodenně používaných technologií pro tvorbu lepších moderních webových aplikací.

Vývojáři používající HTML5 dostávají do rukou nové nástroje k tvorbě lepšího uživatelského rozhraní - a to od popisnějších značek a lepší komunikace mezi weby nebo okny až k animacím a vylepšené podpoře médií [6].

Specifikace HTML5 je složena z několika víceméně nezávislých částí, například:

- nové HTML značky (tagy) sémanticky definující strukturu stránky,
- perzistentní úložiště formou asociativního pole,
- relační databáze s podporou transakcí,
- podpora offline aplikací.

HTML verze 5 se od verze 4 liší novými, zkrácenými a rychlejšími zápisy značek. Autoři dávají důraz na jednoduchost a zároveň účinnost. V HTML5 je též možné vytvořit aplikaci, která funguje v prohlížeči i tehdy, kdy uživatel nemá internetové připojení, a která ukládá data do lokálního úložiště na uživatelově počítači. Je-li internetové připojení k dispozici, může aplikace synchronizovat data se vzdáleným serverem.

HTML5 se také zaměřilo na sémantiku webových stránek a převážně na zvýšení přehlednosti zdrojového kódu. Většina stránek je dnes tvořena obvyklými částmi, jako je hlavička, různé sloupce a patička. Tyto části jsou odlišeny pomocí tagu `div` a jeho atributů `id` nebo `class`. V současné verzi HTML 4 žádné speciální tagy na toto rozlišení nejsou [8].

7.2.1 Canvas

Canvas je plátno, které má svojí výšku a šířku a je použito ke kreslení objektů (čtverců, trojúhelníků, kruhů), přechodů a také k vykreslování textů za běhu webového prohlížeče.[17]

Původně byl představen společností Apple pro použití v operačním systému Mac OS X pro WebKit komponenty, kterými jsou realizovány např. Dashboard widgety (miniaplikace rozmístitelné na pracovní ploše, známé také z Windows Vista a 7) nebo prohlížeč Safari. Později byl implementován do prohlížečů s jádrem Gecko (např. Mozilla Firefox), do prohlížeče Opera a také standardizován WHATWG pro nově navržené specifikace next-gen webových technologií [14].

V HTML5 má canvas svůj tag `<canvas>` je to párový tag, čili je nutno ukončit pomocí `< /canvas>`. Na jedné HTML stránce je možnost mít více canvasů [16]. Je to pouze kontejner pro zobrazení grafických objektů. Veškeré kreslení se provádí skriptovacím jazykem JavaScript [18]. Canvas umožňuje nejen kreslení objektů, ale vlastní také metody pro kreslení čar a vkládání obrázků a poté následující úpravu pixel po pixelu. Tyto metody jsou použity v projektu. Jeho velkou výhodou je, že má neomezenou možnost vytvořit jakýkoliv nástroj či efekt.

```
//pomocí javascriptu hledáme Canvas s identifikátorem "myCanvas"  
var canvas=document.getElementById("myCanvas");  
  
//poté voláme metodu getContext(), kdy nám musí projít řetězec "2d"  
var context=canvas.getContext("2d");  
  
//na plátno si nakreslíme červený obdélník  
ctx.fillStyle="#FF0000";  
ctx.fillRect(0,0,150,75);
```

Tabulka 4: Příklad přístupu do canvasu

7.3 CSS3

Cascading Style Sheets (CSS) již existuje poměrně dlouhou dobu. Současná verze 3 je ve vývoji od roku 2005 a k dnešnímu dni není 100% standardem. Výhodou je, že dnešní prohlížeče se přizpůsobují vývoji a tak stále přidávají podporu, pro právě zmiňované CSS ve verzi 3. Jeho úkolem je oddělit strukturu HTML od stylů. Tomuto programování se říká nevtíravé. Nejlépe je umístit všechny styly do externího souboru, kdy je poté možnost je znova využít.

Vše se řeší pomocí ukazatelů (id, třídy) daného elementu. ID je jedinečný identifikátor, který by se neměl v dokumentu opakovat. Tzv. Classa (třída) nám umožňuje aplikovat styly na více elementů. Např. seznam, které má nějaké položky a právě u nich je potřeba aplikovat vlastnosti jako stejnou velikost a barvu pozadí, jednoduše přiřadí každé položce v seznamu třídu, která reprezentuje dané vlastnosti. Když je seznam obalen do třídy pak lze přes tuto třídu ukázat na dané položky v seznamu, tak aby nebyly ovlivněny ostatní.

Díky těmto ukazatelům se ušetří spoustu času. Pokud je potřeba změnit vzhled stránky, stačí pouze změnit styly v CSS souboru a tím se aplikuje v celém projektu.

CSS3 přináší spoustu vlastností, parametrů a dalších novinek, které se dají nastavit elementům.

Nevýhodou je podpora prohlížečů, která opět narušuje verze Internet Exploreru 8 a nižší. Proto vývoj musí brát zřetel i na uživatele, kteří ho využívají. Styl kódování webu pak vytváří obrázkové weby, které jsou náročné na stahování. Tento styl webu vzniká z důvodu neznalosti či lenosti uživatelů a rozpočtu věnovaný aktualizaci webových prohlížečů ve firmách, kvůli kterým je vytvářen.

7.4 jQuery

jQuery [10] je JavaScriptová knihovna s širokou podporou prohlížečů, která klade důraz na interakci mezi JavaScriptem a HTML. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu.

jQuery je svobodný a otevřený software pod licencí MIT [22].

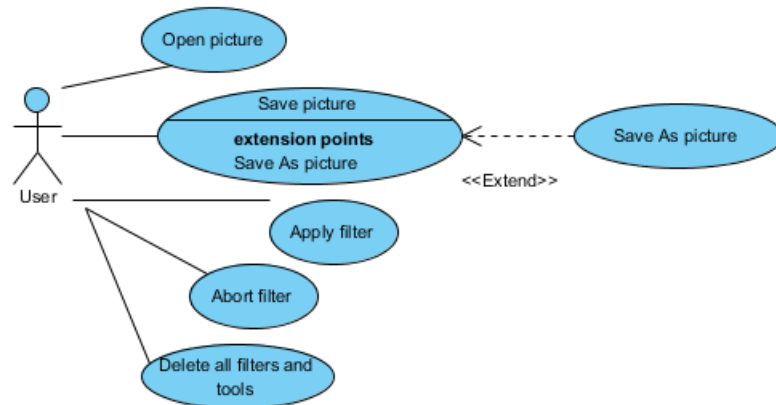
Stejně jako CSS odděluje „zobrazovací“ charakteristiky od struktury HTML, jQuery odděluje „chování“ od struktury HTML. Například místo přímé specifikace on-click události přímo v HTML kódu tlačítka by stránka řízená jQuery napřed našla vhodný element tlačítka a potom změnila jeho manipulátor události. Oddělení chování od struktury se také často nazývá jako princip nevtíravého JavaScriptu [3].

jQuery nabízí následující funkce:

- Výběr DOM elementů pomocí otevřeného cross-browser selektorového engine Sizzle, odnože projektu jQuery
- Funkce pro procházení a změnu DOM (včetně podpory pro 1–3 a základní XPath)
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX
- Rozšiřitelnost
- Utility – např. informace o prohlížeči nebo funkce each
- Javascriptové pluginy

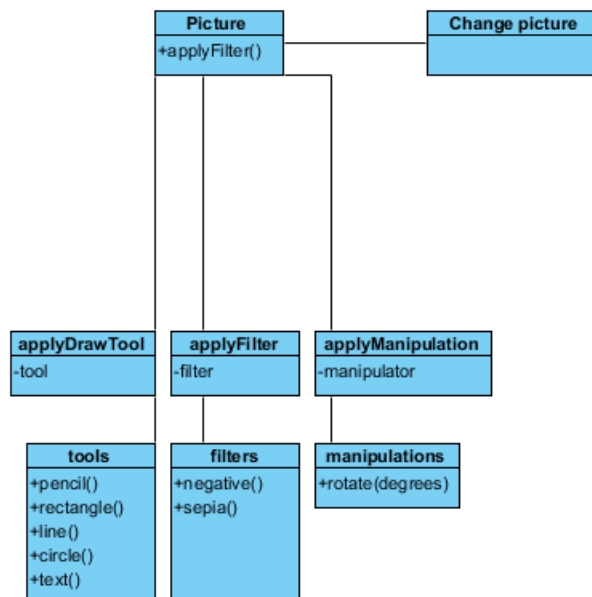
8 Realizace grafického editoru

8.1 USE CASE diagram



Obrázek 2: USE CASE Diagram

8.2 CLASS diagram



Obrázek 3: CLASS Diagram

8.3 Návrh designu

Vytvoření designu vzniklo v profesionálním grafickém programu Adobe Photoshop CS6, zde byly navrženy ikony, boxy a barevná sladěnost. Ikony jsou dány do jednoho jediného souboru "tools.png" a to z toho důvodu, že po načtení stránky se soubor načte do mezipaměti pouze jednou a poté se z něho pouze čte na základě souřadnic (x, y). Umožňuje to jednoduchá vlastnost v CSS3: "background-position".

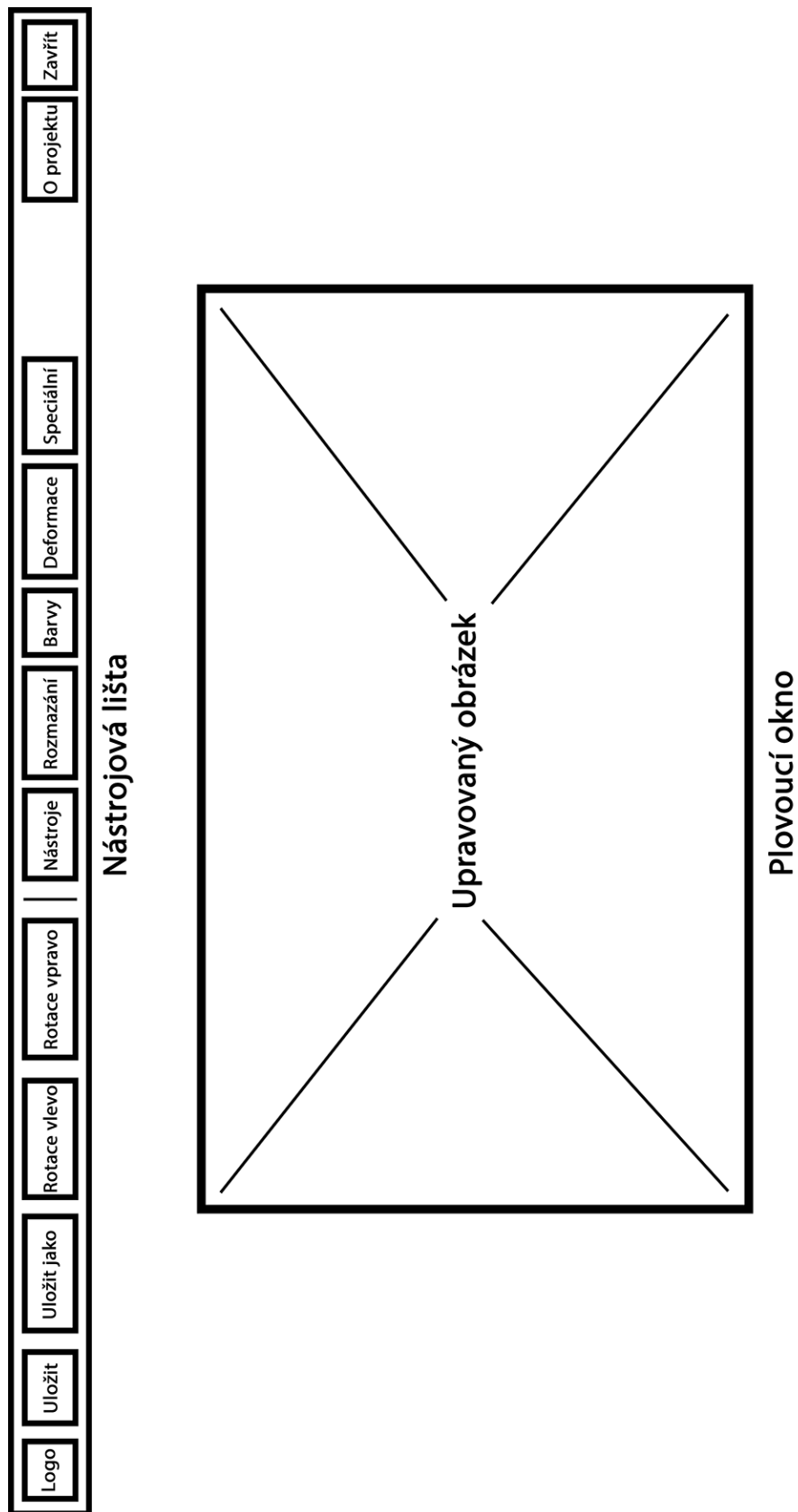
Návrh grafického editoru je realizován pomocí zobrazovacího nástroje Colorbox [21], který usnadňuje další vývoj aplikace. Colorbox byl vytvořen Jackem Moorem. Colorbox obsahuje velké množství funkcí, pro účely této práce jich bylo využito pouze několik. Důvodem je, že JavaScriptová knihovna a soubor se styly obsahují rozsáhlé kódy, tím pádem by se zvětšil datový tok mezi klientem a daným serverem. Zobrazování obrázků pomocí tagu <a>, tj. pomocí odkazu na daný obrázek, se přeneso jeho odkazovaný obrázek (originální) do Colorboxu a poté zařídí jasný import do canvasu pomocí JavaScriptu. Po kliknutí se obrázek uloží do proměnné "url" a poté je možnost importovaný obrázek dále upravovat.

```
//uložení obrázku po kliknutí do proměnné url  
var url=$(this).attr("href");
```

Tabulka 5: Import obrázku do canvasu

Důraz by měl být kladen na kompatibilitu prohlížečů. Podpora canvasu je v dnešní době v následujících prohlížečích: Google Chrome, Firefox, Opera, Safari, Internet Explorer 9 a vyšší. Také existují různé JavaScriptové podpory pro verze Internet Exploreru 8 a nižší, ale funkčnost není zajištěna tak jako u nativní podpory ze strany vývojářů daných prohlížečů. Příkladem podpory pro nižší verze je od Google inc. JavaScriptová knihovna ExCanvas.

Editor je založen na jednoduchosti, přehlednosti a jednoduchém ovládní. Upravovaný obrázek je vycentrován na střed obrazovky a to díky Colorboxu, který má funkce ošetření výpočtů vzdáleností od okrajů. V případě změny rozlišení okna je obrázek opět vycentrován na střed.



Obrázek 4: Wireframe (drátěný model) aplikace

8.4 Programování designu

8.4.1 Menu

Menu je tvořeno pěti položkami, které se dále rozvíjejí do dalších podpoložek. Po kliknutí na položku v menu se rozbalí daný seznam podpoložek a v případě kliknutí na další položku, se předchozí zabalí. Jednotlivé položky mají nastavenou jednu třídu, na kterou reaguje událost click, která zjistí, zda je třída dropdown-container zobrazena (display: block) a pokud ano, zabalí ostatní.

```
//pro každý element s třídou .category
$('div.category').each(function() {
//do proměnné dropdown uložíme element s třídou .category
var $dropdown = $(this);
//element s třídou .menuItem reaguje na událost click
$('div.menuItem', $dropdown).click(function(e) {
e.preventDefault();
//do proměnné $div uložíme obě typy elementů
$div = $('div.menuFilters', $dropdown);
//reakce po kliknutí, menu se zobrazí
$div.slideToggle('fast');
//pokud je kliknuto na jiný element .menuFilters, ostatní skryj
$('div.menuFilters').not($div).slideUp('fast');
//vrať false
return false;
});
});
```

Tabulka 6: Zobrazování a skrývání menu

8.4.2 Problémy při vývoji aplikace

Problém č.1

Kompatibilita Internet Explorer 8 a nižší

Řešení

Ošetření nekompatibility lze vyřešit dvěma způsoby.

Prvním z nich je využití knihovny Excanvas (ExplorerCanvas) podporovanou samotnou společností Google inc.

Druhou možností je využití chybové hlášky zobrazující se v samotném Colorboxu grafického editoru.

Knihovna Excanvas byla otestována s negativními výsledky. Využívané filtry a nástroje nefungují korektně, jak je vyžadováno.

Proto byla zvolena druhá varianta řešení v podobě chybové hlášky, která vyzývá uživatele ke stažení aktuálního webového prohlížeče.

```
<canvas id='myCanvas' width='500' height='500'>  
<p>  
Váš prohlížeč nepodporuje element Canvas.  
Prosíme, stáhněte si Google Chrome či Firefox!  
</p>  
</canvas>
```

Tabulka 7: Varovná hláška při nekompatibilitě prohlížeče

Problém č.2

Colorbox má na sobě nabaleno spoustu dalších elementů, které způsobují rozpad GUI a dále nezobrazuje reálnou velikost canvasu. Tento problém způsobuje zobrazování vertikálního i horizontálního scrollbaru (posuvníku).

Řešení

Při zasažení do JavaScriptové knihovny Colorboxu vznikají další problémy a to z důvodu, že na předpřipravené funkce jsou navázány další a po jejich odstranění Colorbox nefunguje správným způsobem. Je tedy nutné vytvořit vlastní funkci, která zaručí odstranění daných obalových elementů. Tuto možnost zajišťuje vlastní funkce nazvaná `hideBorders()`, která odstraní obalové elementy.

```
//funkce pro odstranění obalových elementů  
$( 'id všech obalových elementů' ).remove();
```

Tabulka 8: Odstranění obal. elementů

8.5 Programování funkcí

Pro výslednou aplikaci jsou naprogramovány následující funkce grafického editoru:

- Načtení obrázku do Colorboxu
- Přizpůsobení funkční části Colorboxu
- Změna rozlišení (včetně zjištění rozlišení aktuálního obrázku)
- Rotace
- Aplikování filtrů

8.5.1 Načtení obrázku do Colorboxu

Colorbox nabízí několik možností načtení obsahu. Pro potřeby aplikace jsou vybrány tři metody: `onComplete`, metoda, která volá funkce po načtení obsahu.

Dále metoda `onClosed`. Slouží k načtení funkcí po zavření `Colorboxu`. Poslední závěrečná metodou je `html`, která načte šablonu do `Colorboxu`.

Po kliknutí na obrázek je jako první zavolána metoda `html`, zobrazující řetězec `html`. Vytvořený řetězec je párový tag `canvas`, ukrývající se ve funkci `createCanvas()`, který ho vsadí do `Colorboxu`.

Další částí kódu je selektor vybírající nástroje, které se mají zobrazit při načtení obrázku do `canvasu`. Způsob načtení obrázku do `canvasu` a způsob získání `contextu` je velmi jednoduchý. Stačí vytvořit jednoduchou proměnnou, která se nazývá `myCanvas` a pomocí selektoru do proměnné uložíme prvek `<canvas>` z `DOMu` [27].

Dalším krokem je vytvoření proměnné `context`, která reprezentuje `context canvasu`. V proměnné je uložen `canvas` se zavolanou metodou `getContext('2D')`, která zajistí v jaké dimenzi se daná aplikace pohybuje. Zvolena je dimenze `2D` zajišťující práci se dvěma osami `X` a `Y`.

Pro zajištění načtení obrázku je vytvořen nový objekt, který reprezentuje obrázek. Dále je nutné zajistit zdroj obrázku. Toho je dosaženo tak, že metoda `src` přiřadí vybraný obrázek, který se má vykreslit. Aby se obrázek vykreslil do `canvasu`, je využito proměnné `context`, která si zavolá metodu `drawImage()` s parametry zdrojového obrázku - počátečního bodu vykreslení (`X`, `Y`) a velikosti obrázku.

```

//získá se odkaz na obrázek, na který se kliklo a dále se získá canvas a jeho
context
var url = $(this).attr("href");
//pomocí javascriptu hledáme Canvas s identifikátorem "myCanvas"
var canvas = document.getElementById("myCanvas");
//poté voláme metodu getContext(), kdy nám musí projít řetězec "2d"
var context = canvas.getContext("2d");
//pomocí javascriptu hledáme Canvas s identifikátorem "myCanvas"
var imageObj = new Image();
//načtení funkcí až po načtení obrázku
imageObj.onload = function() {
//metoda drawImage vykreslí obrázek do canvasu
context.drawImage(imageObj, 0, 0, imageObj.width, imageObj.height)
});
//zdroj obrázku
imageObj.src = url;

```

Tabulka 9: Načítání obrázku do canvasu

8.5.2 Přizpůsobení funkční části Colorboxu

Z Colorboxu jsou vynechány zbytečné části, které zatěžují webový prohlížeč. Ve vývoji aplikace jich není potřeba. Hlavními požadavky na aplikaci jsou: minimální datový tok, efektivita načtení a rychlé zpracování dat. Očesání proběhlo ve zdrojovém souboru jquery.colorbox.js. Další možné úpravy jsou provedeny v css souboru Colorboxu, kdy je upraven jeho vzhled v podobě moderního designu. Úpravy jsou provedeny ve velkém rozsahu, zdrojový kód je zmenšen na minimum a přidán do jediného souboru s ostatními styly. Tímto souborem je editor.css.

8.5.3 Aplikování filtrů

Aplikování filtru je složitý proces, který se dělí na tři části. První částí jsou události, reagující na kliknutí na dané filtry. V druhé části jsou funkce, zajišťující

získání hodnot z posuvníků daného filtru a poté zavolání funkce s parametry na aplikování filtru. Třetí část procesu je již samotná knihovna s filtry, díky ní je možné manipulovat s každým pixelem v canvasu.

```
//událost na kliknutí daného tlačítka
$("#applyBlur").click(function() {
//funkce na zobrazení vyčkávací hlášky
wait();
setTimeout() {
Filters.properties.blur(data, context);
}), 1000;
});
```

Tabulka 10: První část: událost click

```
//událost na kliknutí daného tlačítka
var filterList = function() {
this.blur = function(data, context) {
var amount = $("#blurProperties").val();
JSManipulate.blur.filter(data, {amount: amount});
context.putImageData(data, 0, 0);
});
};
```

Tabulka 11: Druhá část: zjištění hodnot posuvníků a zavolání funkce na aplikování filtru

```
/* Potřebná vstupní data, vezme se pixel po pixelu, aplikuje se na něj filtr a poté vykreslí do canvasu.*/
```

Zdrojový kód je rozsáhlý, k nalezení v archivu projektu po jménem jsmanipulate.js

Tabulka 12: Třetí část: aplikování filtru

8.5.4 Změna rozlišení

Změna rozlišení je nastavena na událost change. Jsou vytvořeny dva vstupy (input type='number'), které přijímají parametry šířku a výšku. Při změně šířky je zavolána funkce canvasWidthChanged na základě události change. V této funkci je provedena konverze z canvasu do obrázku a následně tato proměnná je předána na vykreslení obrázku do canvasu. Stejný postup je i při změně výšky.

```
formElement = document.getElementById("canvasWidth");
formElement.addEventListener('change', canvasWidthChanged, false);
function drawScreen(imageObj) {
  context.drawImage(imageObj, 0, 0, canvas.width, canvas.height);
  $.colorbox.resize({width: canvas.width, height: canvas.height});
}
function canvasWidthChanged(e) {
  var target = e.target;
  var imageObj = convertCanvasToImage(canvas);
  canvas.width = target.value;
  drawScreen(imageObj);
}
```

Tabulka 13: Zjištění hodnot a následné volání funkce na aplikování filtrů

8.5.5 Ořez

Ořez je prováděn na základě dvou canvasů. První canvas je aktuálně vyobrazen a druhý canvas je skryt a slouží k ukládání dočasněho ořezu a poté je vyobrazen jako primární canvas.

Pro funkčnost ořezu jsou potřeba vytvořit události - mouseDown, mouseMove, mouseUp, které interagují s uživatelem. V události mouseDown jsou zjištěny a přiřazeny do proměnných aktuální souřadnice stisku tlačítka na myši. Uchovávají se souřadnice osy x a osy y, kde je aktuálně stisknuto tlačítko na myši. Tento aspekt je důležitý pro další výpočty jako například šířky a výšky daného

ořezu. Dále v události `mouseDown` je vytvořena proměnná vytvářející element `< div >`, který má nastaven absolutní pozici s vlastnostmi `top` a `left`. Jejich hodnoty jsou vypočteny a poté dojde k nastavení ohraničení `border` s parametry tloušťky `1px` stylem čerchované čáry a červené barvy pro lepší viditelnost na obrazovce.

V události `mousemove` jsou vytvořeny proměnné `w` a `h`, které reprezentují šířku a výšku ohraničení. Do těchto proměnných jsou přiřazeny hodnoty, které jsou vypočteny pomocí absolutní hodnoty pozice myši při události `mouseDown` minus aktuální pozice. Vypočítané proměnné `w` (`width = šířka`) a `h` (`height = výška`) a jsou předány jako parametry vlastností `width` a `height` pomocí jQuery metody `css()` našemu `divu`, který je vytvořen v události `mouseDown`.

Ořez `canvasu` probíhá v události `mouseup` (reakce na puštění tlačítka). Do proměnných `endMousePos.x` a `endMousePos.y` jsou uloženy souřadnice bodu, kdy je puštěno tlačítko myši. Poté se vytvoří proměna `pTarget`, kam se přiřadí cíl, kde se nachází výsledný ořez. Cíl výsledného ořezu je odeslán do `color-boxu`. Pro uchování nových informací o ořezu je vytvořena skupina proměnných: `cropStartX`, `cropStartY`, `cropEndX`, `cropEndY`, `cropWidth`, `cropHeight`.

Výpočet proměnné `cropStartX` je dán tak, že je vzata souřadnice `x` při události `mouseDown` (stisk tlačítka na myši) a následně je od ní odečtena pozice levého okraje `canvasu`.

```
var cropStartX = startMousePos.x - pTarget.position().left;
```

Tabulka 14: Výpočet hodnoty `cropStartX`

Výpočet proměnné `cropStartY` je dán tak, že je vzata souřadnice `y` při události `mouseDown` (stisk tlačítka na myši) a následně je od ní odečtena pozice horního okraje `canvasu`.

```
var cropStartY = startMousePos.y - pTarget.position().top;
```

Tabulka 15: Výpočet hodnoty `cropStartY`

Výpočet proměnné `cropEndX` je dán tak, že je vzata souřadnice `x` při události `mouseUp` (upuštění tlačítka na myši) a následně je od ní odečtena pozice levého okraje canvasu.

```
var cropEndX = endMousePos.x - pTarget.position().left;
```

Tabulka 16: Výpočet hodnoty `cropEndX`

Výpočet proměnné `cropEndY` je dán tak, že je vzata souřadnice `y` při události `mouseUp` (upuštění tlačítka na myši) a následně je od ní odečtena pozice horního okraje canvasu.

```
var cropEndY = endMousePos.y - pTarget.position().top;
```

Tabulka 17: Výpočet hodnoty `cropEndY`

Výpočet proměnné `cropWidth` je dán tak, že je vzata absolutní hodnota počáteční souřadnice při události `mousedown` (stisk tlačítka na myši) a následně je od ní odečtena počáteční souřadnice `X` při události `mouseUp`.

```
var cropWidth = Math.abs(startMousePos.x - endMousePos.x);
```

Tabulka 18: Výpočet hodnoty `cropWidth`

Výpočet proměnné `cropHeight` je dán tak, že je vzata absolutní hodnota konečné souřadnice při události `mousedown` (stisk tlačítka na myši) a následně je od ní odečtena konečná souřadnice `Y` při události `mouseUp`.

```
var cropHeight = Math.abs(endMousePos.y - endMousePos.y);
```

Tabulka 19: Výpočet hodnoty `cropHeight`

Na základě všech vypočtených hodnot přichází na řadu funkce `cropImage`. V jejím nitru se nachází kód vytvářející dočasný canvas, ke kterému poté získá `context`, díky němuž je možné natáhnout obsah zdrojového canvasu do dočasného.

```
var tempCanvas = $(' < canvas >< /canvas >')[0];
var tempCanvasContext = tempCanvas.getContext('2d');
```

Tabulka 20: Vytvoření dočasného canvasu a získání jeho contextu

Dále je potřeba nastavit dočasnému canvasu šířku a výšku, které jsou získány z primárního canvasu a poté je vkreslen oříznutý primární canvas do dočasného.

```
$(tempCanvas).attr({width: $(canvas).attr('width'),
height: $(canvas).attr('height')});
tempCanvasContext.drawImage(canvas, 0, 0);
```

Tabulka 21: Nastavení šířky a výšky dočasnému canvasu

V další části pro finální vykreslení obrázku je potřeba nastavit primárnímu canvasu šířku a výšku z obrázku, která je převzata z parametrů funkce.

```
$(canvas).attr({width: imgWidth, height: imgHeight});
canvas.getContext("2d").drawImage(tempCanvas, startX, startY, imgWidth,
imgHeight, 0, 0, imgWidth, imgHeight);
```

Tabulka 22: Nastavení šířky a výšky primárního canvasu a finální vykreslení

8.5.6 Rotace

Pro použití rotace v canvasu je nutné mít pět aspektů pro její vykonání. Canvas, jeho context, úhel, šířku a výšku canvasu. Do proměnných *w* a *h* je nahrána šířka a výška canvasu (Pozor! Je nutné obrátit pořadí. Do proměnné *w* je přiřazena výška a do proměnné *h* šířka). Pomocí contextu javascript umožní použít metodu `translate()`, díky ní je možné přemisťovat obrázek. Metodě `translate` jsou předány dva parametry pro umístění na ose *x* a *y*.

Rotace ve směru hodinových ručiček je provedena tak, že za parametr *x* je dosazena proměnná *h*, která reprezentuje výšku canvasu a za parametr *y* je dosazena 0. Pokud je prováděna rotace proti směru hodinových ručiček, za parametr *x* je dosazena 0 a za parametr *y* proměnná *w*, která reprezentuje šířku canvasu. Další potřebnou metodou contextu je metoda `rotate()`, která přijímá parametr úhel.

(Pozor! Nutné použít radiány, nikoli stupně). V poslední řadě zbývá vykreslit výsledek do canvasu. O toto se stará metoda `drawImage()`.

Připravená funkce, která je poté volána událostí `click`, jsou jí předány čtyři parametry - `canvas`, `context`, šířka a výška aktuálního obrázku v canvasu. V závěru funkce je zavolána metoda `detectSizeCanvas()`, která zjistí velikost canvasu a nastaví je na vstupy.

```
canvas.width = h;  
canvas.height = w;  
canvas.translate(h, 0);  
canvas.rotate(Math.PI / 2);  
canvas.drawImage(image, 0, 0);
```

Tabulka 23: Rotace canvasu ve směru hodinových ručiček

```
canvas.width = h;  
canvas.height = w;  
canvas.translate(0, w);  
canvas.rotate(-(Math.PI / 2));  
canvas.drawImage(image, 0, 0);
```

Tabulka 24: Rotace canvasu proti směru hodinových ručiček

9 Testování

Testování grafického editoru probíhalo při postupném vývoji, kdy každá fáze byla vystavena k otestování uživateli.

Během testovací doby byly využity nástroje Firebug pro Firefox a výpisy chybových hlášek přímo z daných webových prohlížečů.

Testování kompatibility bylo provedeno v prohlížečích Google Chrome, Mozilla Firefox, Opera a Internet Explorer (9, 10, 11)

Zjištěné chyby a různé nedostatky budou následně vyloučeny v budoucí aktualizaci grafického editoru.

Testovací scénář:

- Otevření obrázku v Colorboxu
- Aplikování jednotlivých filtrů
- Změna rozlišení obrázku
- Rotace obrázku
- Ořez
- Následné aplikování obrázku
- Následné změna obrázku

10 Závěr

V rámci bakalářské práce byl vytvořen grafický editor, který umožňuje provádět jednoduché úpravy obrázků před odesláním na server.

Byla provedena analýza offline grafických editorů na základě komparace, která přinesla výsledky v podobě přehledu filtrů a nástrojů, které offline grafické editory obsahují.

K dosažení cíle práce je potřeba dobře znát a ovládat značkovací jazyk HTML (konstrukce, vlastnosti elementů, a pod.), CSS (kaskádové styly, dědění stylů pro usnadnění práce), skriptovací jazyk JavaScript a framework jQuery, který je na tomto jazyku postaven a chování canvasu (vykreslování či překreslování jednotlivými contexty).

Pro jednoduchou demonstraci byla vytvořena základní šablona s obrázky, na kterých je možno provádět operace aplikování filtrů a nástrojů.

Grafický editor FastPicture obsahuje sadu 40 filtrů a čtyř nástrojů (rotace ve směru hodinových ručiček a proti směru hodinových ručiček, změna rozlišení, ořez), které dostačují pro jednoduchou korekturu obrázků.

Bakalářskou prací je demonstrováno využití nových technologií HTML5 v praktickém zaměření, na jejich základě byl vytvořen Grafický editor FastPicture, který ovládá řadu propracovaných filtrů a jednoduchých nástrojů pro úpravu obrázků. Návrh tohoto editoru vzešel z moderních technologií - framework jQuery, Colorbox. Design byl navrhnout v grafickém editoru Adobe Photoshop CS6, kde byl vytvořen wireframe (drátový model). Následné nasazení grafiky se podobá designu Modern UI navržené firmou Microsoft, která se nachází v současných systémech Windows 8.1. Naprogramované funkce v Javascriptu a frameworku jQuery jsou rozděleny každé do svých souborů pojmenovaných podle své funkce, kvůli lepší orientaci v případě rozšíření a dalších budoucích oprav. Hlavním souborem je skript function.js řídící celý proces načtení obrázku do canvasu a poté si volá další soubory s funkcemi, které uživatel zvolí v rozhraní aplikace.

Reference

- [1] DANNHOFEROVÁ, Jana. Velká kniha barev: kompletní průvodce pro grafiky, fotografy a designéry. 1. vyd. Brno: Computer Press, 2012, 352 s. ISBN 978-80-251-3785-7.
- [2] JQuery: kuchařka programátora. Vyd. 1. Brno: Computer Press, 2010, 436 s. ISBN 978-80-251-3152-7.
- [3] WIKIPEDIA. JQuery [online]. 22. 4. 2013 [cit. 2013-07-02]. Dostupné z: <http://cs.wikipedia.org/wiki/JQuery>
- [4] Html5rocks: WHY HTML5 ROCKS [online]. 2011 [cit. 2013-07-02]. Dostupné z: <http://www.html5rocks.com/en/why>
- [5] CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.
- [6] HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Vyd. 1. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3576-1.
- [7] LUBBERS, Peter, Brian ALBERS a Frank SALIM. HTML5: programujeme moderní webové aplikace. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6.
- [8] W3C EDITOR'S DRAFT. HTML5 [online]. 2013 [cit. 2013-07-03]. Dostupné z: <http://www.w3.org/html/wg/drafts/html/CR/>
- [9] W3SCHOOLS ONLINE WEB TUTORIALS. W3Schools Online Web Tutorials: the world's largest web development site [online]. 1999, 2013 [cit. 2013-01-19]. Dostupné z: <http://www.w3schools.com/>
- [10] THE JQUERY FOUNDATION. The jQuery foundation: The Write Less, Do More, JavaScript Library [online]. 2012 [cit. 2013-01-19]. Dostupné z: <http://jquery.com/>

- [11] FREE SOFTWARE FOUNDATION. GNU AFFERO GENERAL PUBLIC LICENSE [online]. 19.12. 2007 [cit. 2013-06-24]. Dostupné z: <http://www.gnu.org/licenses/agpl-3.0.html>
- [12] W3C RECOMMENDATION. Media Queries [online]. 19.6. 2012 [cit. 2013-06-24]. Dostupné z: <http://www.w3.org/TR/css3-mediaqueries/>
- [13] A LIST APART and OUR AUTHORS. A LIST APART: Responsive Web Design [online]. 2013 [cit. 2013-06-24]. Dostupné z: <http://alistapart.com/article/responsive-web-design>
- [14] WIKIPEDIA. Canvas [online]. 25.3 2013 [cit. 2013-07-03]. Dostupné z: <http://cs.wikipedia.org/wiki/Canvas>
- [15] FULTON, Steve a Jeff FULTON. HTML5 Canvas: native interactivity and animation for the Web. 1st ed. Sebastopol, CA: O'Reilly, 2011, xix, 628 p. ISBN 14-493-9390-X.
- [16] W3SCHOOLS. HTML5 Canvas [online]. 2011 [cit. 2013-07-02]. Dostupné z: http://www.w3schools.com/html/html5_canvas.asp
- [17] ZDROJAK.CZ. : říkejme tomu plocha na kreslení [online]. Jiří Sekera, 23.3.2012 [cit. 2013-07-02]. Dostupné z: <http://www.zdrojak.cz/clanky/canvas-rikejme-tomu-plocha-na-kresleni/>
- [18] W3Schools: JavaScript Tutorial [online]. 2011 [cit. 2013-07-02]. Dostupné z: <http://www.w3schools.com/js/default.asp>
- [19] WIKISKRIPTA, projekt sítě lékařských fakult MEFANET. Poruchy barvocitu [online]. 2011 [cit. 2013-08-11]. Dostupné z: http://www.wikiskripta.eu/index.php/Poruchy_barvocitu
- [20] CBS INTERACTIVE INC. Download.com: C—Net [online]. 2001 [cit. 2013-08-11]. Dostupné z: <http://download.cnet.com/windows/?tag=hdr>

- [21] ColorBox [online]. Jack Moore. 2012 [cit. 2013-09-07]. Dostupné z:
<http://www.jacklmoore.com/colorbox/>
- [22] OPEN SOURCE INITIATIVE. The MIT License (MIT) [online].
2013 [cit. 2013-11-19]. Dostupné z:
<http://opensource.org/licenses/MIT>
- [23] W3SCHOOLS. CSS3 Reference [online]. 2011 [cit. 2013-11-19].
Dostupné z:
http://www.w3schools.com/cssref/css3_browser_support.asp
- [24] GOOGLE INC. Google+ Online graphic editor [online]. 2013 [cit.
2013-11-19]. Dostupné z:
<https://support.google.com/plus/answer/1053729>
- [25] ADOBE SYSTEMS INCORPORATED. Photoshop Online Tools —
Photoshop.com [online]. 2012 [cit. 2013-11-19]. Dostupné z:
<http://www.photoshop.com/tools?wf=editor>
- [26] AUTODESK. Online image editor pixlr free - fix photos direct in
your browser [online]. 2012 [cit. 2013-11-19]. Dostupné z:
<http://pixlr.com/express/>
- [27] Document Object Model. W3C. Document Object Model [online].
2005 [cit. 2013-11-19]. Dostupné z: <http://www.w3.org/DOM/>
- [28] MEDIACENTRIK. CMS [online]. 2007 [cit. 2013-12-06]. Dostupné z:
<http://www.mediacentrik.cz/cms.dic>
- [29] Grafické filtry & zobrazovací jednotky. ČVUT Praha, 2007. Dostupné
z: <http://fyzsem.fjfi.cvut.cz/2007-2008/Zima07/proc/graffiltry.pdf>
- [30] *Implementace grafických filtrů pro zpracování rastrového obrazu.*
Masarykova univerzita, 2008. Dostupné z:
http://is.muni.cz/th/60754/fi_b/bakalarka.pdf

A Aplikace

Aplikace se nachází na CD přiložené k této práci.

B Postup instalace

- Rozbalte soubor FastPicture.zip do Vašeho projektu.
- Následně ve Vašem hlavním html nebo php souboru s názvem index připojte následující knihovny (také součástí zip archivu)
 - jQuery knihovna - v archivu pod názvem jquery.min.js nebo také ke stažení na oficiálních stránkách [jQuery.com](http://jquery.com)
 - FastPicture editor - v archivu pod názvem functions.js (veškeré načtení zbylých knihoven provede samotná funkce appendScripts)
 - Nakopírovat soubor loading.gif do svého projektu, cestu lze změnit v souboru editor.css
- FastPicture editor reaguje na následný zápis důležitý pro zobrazení obrázku. ``