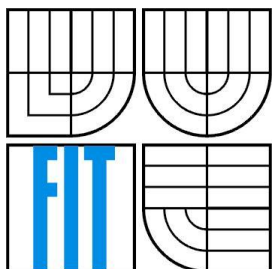


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SIMULÁTOR A TRENAŽÉR DIGITÁLNÍCH FOTOAPARÁTŮ ŘADY CANON A70

SIMULATOR AND TRAINER DIGITAL CAMERAS SERIES OF CANON A70

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Daniel Řepa

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. Jiří Kunovský , CSc.

BRNO 2009

Zadání bakalářské práce

Řešitel: **Řepa Daniel**

Obor: Informační technologie

Téma: **Simulátor a trenažer digitálních fotoaparátů řady Canon A70**

Kategorie: Modelování a simulace

Pokyny:

1. Seznamte se podrobně s obsluhou fotoaparátů řady Canon A70. Vytvořte program pro prezentaci názorných fotografií, kterými budete demonstrovat, jaké možnosti poskytuje digitální fotoaparát v režimu "Prohlížení". Fotografie, včetně odpovídajícího popisu displeje fotoaparátu, vytvořte ve vhodném formátu.
2. Vytvořte podobně program pro prezentaci názorných fotografií v režimu "Fotografování", opět včetně ikon displeje fotoaparátu.
3. Navrhněte a implementujte program, který bude simulovat chování fotoaparátu v režimu "Fotografování" a který bude sloužit jako trenažer.
4. Zhodnoťte výhody a nevýhody navrhovaného řešení.

Literatura:

- <http://www.fotografovani.cz/zaklady/index.html>
- <http://www.paladix.cz/>
- <http://www.digineff.cz/>
- Návod k obsluze fotoaparátu Canon A70

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kunovský Jiří, doc. Ing., CSc., UITS FIT VUT**

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
602 00 Brno, Bežová 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Bakalářská práce se zabývá tvorbou simulátoru a trenažeru fotoaparátu Canon A70 a jeho manuálu. V textu práce je popsána problematika analýzy aplikace, volby technologie a realizace samotné aplikace. V závěrečné části je srovnání s ostatními bakalářskými pracemi na toto téma a anketa ukazující hodnocení aplikace veřejností.

Klíčová slova

Fotoaparát, Canon A70, simulátor a trenažér, javascript, prezentace

Abstract

This bachelor's thesis details the creation of a simulator, a training application and a manual for the Canon A70 camera. The text describes the application analysis, choice of technology and implementation of the simulator. The conclusion compares this bachelor's thesis with others and contains the results of a public inquiry about how well is the application received by users.

Keywords

camera, Canon A70, simulator and trainer, javascript, presentation

Citace

Daniel Řepa: Simulátor a trenažér digitálních fotoaparátů řady Canon A70. Brno, 2009, bakalářská práce, FIT VUT v Brně.

Simulátor a тренаžér digitálních fotoaparátů řady

Canon A70

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Ing. Jiřího Kunovského Csc.

Další informace mi poskytl Bc. Ondřej Melkes.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Daniel Řepa
30.7.2009

Poděkování

Děkuji doc. Ing. Kunovskému za ochotu při vedení bakalářskou prací, cenné informace a poskytnutí ostatních bakalářských prací na toto téma, jejichž myšlenky obohatily tuto práci.

© Daniel Řepa, 2009.

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

Obsah	1
Úvod	3
2 Digitální fotoaparáty	4
2.1 Terminologie.....	4
3 Simulátor a trenažér	7
3.1 Analýza.....	7
3.1.1 Požadavky na aplikaci	7
3.2 Návrh	7
3.2.1 Návrh nástroje na tvorbu simulátorů	8
3.3 Volba prezentační technologie.....	9
3.4 Offline technologie	9
3.5 Online technologie.....	9
3.5.1 Flash.....	9
3.5.2 Java	10
3.5.3 Kombinace html, css, javascript	10
3.6 Volba technologie pro uchování dat	12
3.6.1 Databáze	12
3.6.2 XML	12
3.7 Návrh dokumentu aplikace	13
3.7.1 První návrh.....	13
3.7.2 Druhý návrh	13
3.7.2.1 Kontext	13
3.7.2.2 Mód prohlížení nafocených fotografií	14
3.7.2.3 Menu nastavení fotoaparátu.....	14
3.7.2.4 Režimy focení.....	15
3.7.2.5 Funkce fotoaparátu	16
3.8 Implementace.....	18
3.8.1 Modulární vývoj	18
3.8.1.1 Modul menu nastavení fotoaparátu.....	18
3.8.1.2 Modul nastavení funkcí pro fotografování	19
3.8.1.3 Modul změny režimu	19
3.8.1.4 Modul procházení nafocených fotografií.....	19
3.8.1.5 Modul zoom.....	20
3.8.1.6 Modul změny grafického téma	20

3.8.1.7	Modul jazykové lokalizace	20
3.8.1.8	Modul ovládání aplikace.....	20
3.8.2	Práce s datovou vrstvou	21
3.8.2.1	JSON.....	21
3.8.3	Sloučení modulů	21
3.9	Dodržení zamýšleného postupu	22
4	Manuál	23
4.1	Požadavky.....	23
4.2	Návrh	23
4.3	Provedení	23
5	Ostatní bakalářské práce na téma simulátor a trenžér digitálních fotoaparátů.....	24
5.1	Manuály	24
5.2	Simulátory.....	24
5.2.1	Zaměření	25
5.2.2	Společné myšlenky a postupy	25
5.2.3	Různé nápady.....	25
5.2.4	Odlišné práce	25
5.3	Moje bakalářská práce	25
5.3.1	Volba implementačního nástroje	26
5.3.2	Manuál a simulace	26
5.3.3	Kontextová nápověda pro tlačítka	26
5.4	Zhodnocení a další pokračování tématu	26
6	Možná rozšíření	27
6.1	Zrychlení aplikace	27
6.2	Znalostní báze.....	27
7	Závěr	28
	Literatura	29
	Seznam příloh	30

Úvod

Už dlouho dobu si lidé chtějí udržovat záznamy okamžiků. K tomu slouží fotografie. Ta se vyvíjela dlouhou dobu. Od černobílých přes barevné analogové fotografie až po dnešní digitální. Výhoda digitálních fotografií oproti dřívějším analogovým je v rychlosti jejich pořízení. Principy fotografování i technologie jsou u dnešních analogových a digitálních fotoaparátů téměř totožné, ovšem pro potřeby běžného uživatele, například na dovolené, má digitální fotoaparát nespornou výhodu v jednoduchosti. Digitální fotografii je snadné vyfotit i vymazat. Navíc počet pořízených fotografií, který není limitován počtem snímků kinofilmu, ale velikostí paměťové karty. Jelikož k dnešním fotoaparátům jsou dodávány manuály, které jsou zbytečně složité pro lidi bez technického vzdělání nebo zkušeností, měla by tato práce usnadnit seznámení s digitálními fotoaparáty a zejména fotoaparátem Canon A70.

V první části této zprávy se budu zabývat teoretickými znalostmi pojmů z oblasti digitálních fotoaparátů a základním popisem fotoaparátu Canon A70.

V další části bude popsána analýza aplikace simulátoru, návrh jejího řešení a problémů s kterými se budu potýkat v rámci jeho realizace. Budou zde popsány uvažované technologie a srovnání jednotlivých technologií pro tvorbu aplikace. Následně zde popíši průběh realizace a srovnání mezi zamýšleným řešením, a tím co bude nakonec použito.

V třetí části se budu zabývat tvorbou manuálu k fotoaparátu, jejich návrhem a zhodnocením jeho kladů a záporů.

V poslední části se budu zabývat zhodnocením práce s ostatními bakalářskými pracemi na toto téma a srovnáním použití některých funkcionalit.

Závěrem se zmíním o možnostech rozšíření, která již nebudou implementována, ale mohou mít potenciál pro tento projekt.

2 Digitální fotoaparáty

V dnešní době je na trhu velké množství digitálních fotoaparátů od různých výrobců. Tyto fotoaparáty mají oproti analogovým protějškům několik výhod. Za hlavní výhodu považují možnost rychle pořídit sadu fotografií, které si může uživatel ihned prohlédnout a nechtěné smazat. Další výhodou je rychlá práce s digitálními fotografiemi. Dají se lehce nahrát do počítače a potom upravit v programu, který je k tomu určený. Dále je můžeme snadno umístit na nějaké stránky či do fotogalerie. Analogové fotografie je třeba nejprve vyvolat, potom nasnímat do počítače a až v tomto okamžiku s nimi můžeme pracovat stejným způsobem jako s digitální fotografií. Nevýhodou je menší kvalita záznamu, kterou digitální fotografie má o něco horší než analogová. Dražší digitální fotoaparáty mají již velmi slušné výsledky, ovšem pro běžného uživatele, který chce mít pouze několik památek na dovolenou či výlet do přírody, bohatě postačí běžné přístroje.

2.1 Terminologie

U digitálních fotoaparátů se používá mnoho specifických termínů. Tyto termíny jsou běžné a narazíme na ně ve většině publikací zabývajících se fotoaparáty či focením. Jelikož spousta z těchto pojmů nemá český ekvivalent, je třeba zde uvést alespoň jejich popis. Bez jejich znalosti by nebylo možné zhotovit manuálovou část práce, kde se budeme zabývat popisem jednotlivých funkcionalit. Pro běžnou práci s fotoaparátem stačí několik základních pojmů. Zde jsem informace čerpal z internetových zdrojů, především odkazů [1] a [2].

Automatický režim

Fotoaparát si sám nastaví parametry pro focení, jako jsou zaostření, použití blesku, doba expozice a další. To umožňuje vnitřní software fotoaparátu, který. Tento způsob je ideální pro běžné focení nebo rychlé zachycení nějaké události kdy nemáme čas na nastavování fotoaparátu. Nevýhodou jsou méně kvalitní fotografie.

Manuální režim

Uživatel má možnost nastavit všechny vlastnosti, které ovlivní výslednou fotografii. Nemusí jít vždy o dokonale ostrý obraz, ale třeba i o dosažení tvůrčího záměru. Nevýhodou tohoto způsobu je nutnost nastavit vlastnosti pro každou fotografii zvlášť a hlavně znát všechny parametry jež je třeba nastavit. Pokud se ovšem uživatel s tímto způsobem focení naučí, výsledek je vždy lepší než při automatickém režimu.

Autofocus

Jedná se o systém pro automatické zaostření digitální fotografie i mimo střed hledáčku. Tento systém je používán hlavně u automatického režimu. Pro pořizování kvalitních fotografií není vhodný, jelikož se na něj nedá úplně spolehnout a je lepší věřit vlastním schopnostem.

Doba expozice

Určuje, jak dlouho působí světlo na fotocitlivou vrstvu. Doba expozice je řízená záklopkou a její hodnoty jsou v řádech zlomků sekund. Čím déle na tuto vrstvu světlo působí, tím více světla zachytí a fotografie je jasnější. V opačném případě je fotografie tmavší, proto tato doba musí být vhodně zvolena vzhledem k prostředí a fotografovanému předmětu.

Blesk

Toto zařízení se používá pro osvětlení tmavých prostor. Umožňuje kratší dobu expozice než by jinak bylo třeba. Při fotografování s pomocí blesku může dojít k dvěma nepříjemným jevům a to přexponování fotografie, kdy je příliš dlouhá doba expozice a fotocitlivá vrstva zachytí příliš mnoho světla. Druhá nepříjemná vlastnost je tzv. efekt červených očí.

Clona

Clona řídí množství světla, jež dopadá na fotocitlivou vrstvu. Je tvořena sadou neprůhledných lamel a dle nastavení velikosti otvoru propouští množství světla do fotoaparátu. Při malém množství světla je ostře viditelný pouze předmět, na který je fotoaparát zaostřen, a vzdálenější předměty jsou rozostřené. Při větším množství propouštěného světla budou předměty na výsledné fotografii ostřejší. Clona funguje podobně jako oční duhovka.

Makrofotografie

Je způsob fotografování předmětů na malé vzdálenosti, přibližně do 30cm. U moderních fotoaparátů k tomuto účelu slouží **Makro režim**. Ten umožňuje přiblížit detaily malých objektů jako hmyz, květiny apod. Standardně je možné zvětšení až 30:1.

Objektiv

Optická soustava, která tvoří podstatnou část optických přístrojů, vytváří co nejméně zkreslený reálný obraz pozorovaného předmětu.

Zoom

Označení pro zvětšování snímků pomocí elektronicky ovládaných čoček a prodlužování objektivu. Tento zoom se nazývá optický a nijak neničí kvalitu výsledného snímku. Dále je možno

používat digitální zoom, který je prováděn interním programovým vybavením fotoaparátu a jeho použití ničí kvalitu fotografie. Proto se používání digitálního zoomu nedoporučuje.

ISO citlivost

Vyjadřuje světelnou citlivost snímku. S nižší hodnotou ISO je potřeba kvalitnější osvětlení. Malá hodnota se používá za jasného dne nebo v dostatečně osvětlené místnosti. Vyšší hodnotu ISO použijeme při focení ve tmě nebo šeru, jelikož je snímek citlivější na světlo. Trpí tím ovšem kvalita fotografie, jelikož je patrná „zrnitost“.

3 Simulátor a trenažér

V této části se budeme zabývat simulací samotného fotoaparátu. Nejprve je nutné zanalyzovat požadavky na aplikaci a zvolit správný postup při návrhu aplikace. Výběr vhodné technologie nám výrazně usnadní práci. Nebude to ovšem lehký úkol, jelikož se jich nabízí velké množství. Výrobci k fotoaparátům dodávají pouze papírové manuály, které nebývají příliš srozumitelné, a člověk nemá možnost si fotoaparát řádně prozkoumat a odzkoušet. Tato nevýhoda může potenciálního zájemce od koupě fotoaparátu odradit, jelikož bez potřebných znalostí nebude vědět, zda přístroj poskytuje, co od něj očekává, nebo zda mu bude rozumět. K tomu by měla sloužit tato aplikace, kde si uživatel bude moci odzkoušet základní ovládání fotoaparátu a seznámit se s jeho možnostmi a s těmito znalostmi se již bude moci rozhodnout, zda mu takovýto přístroj vyhovuje.

3.1 Analýza

Musíme si uvědomit, co vše budeme od simulátoru očekávat, co by měl zvládat a umožňovat, komu bude sloužit a jakým způsobem bude informace podávat. Seznámil jsem se s ostatními pracemi na toto téma a mám možnost do analýzy zahrnout i další faktory. Aplikace by měla být uživatelsky příjemná a zároveň dostatečně přehledná tak, aby poskytovala informace co možná nejefektivnějším způsobem.

3.1.1 Požadavky na aplikaci

Jelikož by tato práce měla sloužit pro výuku, musíme ji navrhnout jako studijní materiál či pomůcku. Musí být dostatečně přehledná a v každém kroku obsahovat informace o tom co uživatel dělá nebo má možnost dělat. Tento požadavek by měl být uzpůsoben hlavně studentům univerzity třetího věku, kteří nemají mnoho zkušeností s používáním moderních technologií, a nedostatečně dokumentované ovládání by je mohlo mást a brzy by se v něm ztratili.

Za další podstatnou funkčnost považuji možnost vícejazyčnosti aplikace. Jelikož by tato aplikace mohla být použita v zahraničí nebo pro zahraniční studenty, je důležité, aby byla v jazyce, kterému rozumí, případně by uživatel měl mít možnost si zvolit, v jakém jazyce si chce aplikaci prohlížet.

U ostatních prací na podobné téma vyplynula potřeba snadné dostupnosti této aplikace.

3.2 Návrh

Aplikace by se měla skládat z několika částí:

- Ovládací část – zde se bude jednat o propojení ovládacích prvků s chováním celé simulace. Musíme každé tlačítko přizpůsobit, aby se chovalo správně v daném kontextu, protože některá tlačítka mají například jinou funkčnost v modu prohlížení a jinou v modu focení.
- Grafická část - zde budeme reagovat na podněty od ovládací části, aby se zobrazovala správná data. Taktéž do této části bude patřit vytvoření grafiky pro celou aplikaci.
- Datová část – tato část bude základním prvkem celé aplikace. Bude obsahovat všechny informace, jež mají být zobrazovány a zároveň držet uložená nastavení a kontext aplikace.

K samotné simulaci fotoaparátu by bylo vhodné připojit funkcionalitu pro změnu jazykové mutace aplikace. Pro lepší prezentaci funkcí fotoaparátu bychom měli poskytovat možnost si zvolit fotografii na pozadí z předem připravených snímků, na kterých by uživatel mohl sledovat změny, které jsou způsobeny aktuálním nastavením. Například blesk se na fotce za denního světla neprojeví tak jako při focení na tmavém místě.

3.2.1 Návrh nástroje na tvorbu simulátorů

Díky možnosti seznámit se s ostatními pracemi na toto téma jsem zjistil, že všechny simulátory vycházejí z podobného vzoru. Ač je každý simulátor odlišný dle typu fotoaparátu a jeho funkcí, všechny jsou si podobné z hlediska výsledku. Základní funkce jako zoom či zvolení režimu focení má každý z těchto fotoaparátů. Přesto byl u každého z těchto simulátorů psán zdrojový kód od začátku a žádná jejich část neumožňuje, aby se dala znovupoužít pro simulátor podobného fotoaparátu. Tento styl psaní simulátorů je neefektivní, protože pro dva fotoaparáty, například Canon A70 a Canon A75, které se liší jen nepatrně, by bylo nutné psát dva různé simulátory. Proto jsem přišel s nápadem vytvořit nástroj pro tvorbu takových simulátorů, kde by se dal dynamicky definovat obsah a grafika aniž by se muselo zasáhnout fyzicky do výkonného jádra aplikace. K tomu musíme prostudovat více fotoaparátů. I když jsou si všechny velmi podobné, liší se v ovládání a v možnostech nastavení jak focení, tak samotného fotoaparátu. Taktéž by tento nástroj měl umožnit změnu informativních textů k jednotlivým funkcím fotoaparátu, aniž by bylo potřeba sahat do jádra aplikace.

Z předešlého návrhu aplikace plyne požadavek, aby tento nástroj vycházel z návrhového vzoru model-pohled-řadič (MVC – z anglického názvu Model-View-Controller). Tento systém umožňuje oddělit od sebe grafiku a data tak, aby na sobě byly nezávislé, a komunikace je zprostředkována přes řadič. Tím by se docílilo při dobře navržené struktuře datové vrstvy požadovaného stavu, a to upravovat data reprezentující fotoaparát nezávisle na zbytku aplikace.

V tomto nástroji vytvoříme samotný simulátor fotoaparátu a zároveň budeme mít možnost patřičnou úpravou simulovat fotoaparát s podobnou architekturou.

3.3 Volba prezentační technologie

Nyní je třeba zvolit jazyk, který bude použit pro výkonné jádro aplikace. Technologií, jež se nabízí, je spousta. Hlavní rozdělení těchto technologií je do dvou kategorií. Jednu reprezentují technologie, které se dají používat přes internet a druhou zastupují ostatní, které mohou být spouštěny pouze na počítači po předešlé instalaci. U některých technologií je také potřeba vzít v úvahu náročnost na výkon počítače nebo zatížení síťového provozu, který by výrazně ovlivnil plynulost simulace.

3.4 Offline technologie

Tyto technologie mají výhodu v propracovaných knihovnách pro práci s grafikou. Před tím než je můžeme někde používat, je třeba aplikace zkompileovat. Takto zkompileované aplikace je někdy potřeba nainstalovat na cílový počítač kde bude prezentace spuštěna. Tato varianta mi nepřijde příliš vhodná, jelikož běžný uživatel nemusí mít na počítači práva pro instalaci nových programů. Taktéž mi přijde nevhodný způsob jednotlivých instalací, protože v případě vydání nové verze aplikace, přidání funkcionality nebo opravení nějaké chyby by bylo třeba aplikaci znovu nainstalovat, aby se změny projevíly. Je sice možnost mít nějaký program, který by zajišťoval takovou funkci, ale byla by to zbytečná práce navíc.

3.5 Online technologie

Tyto technologie mají výhodu v možnosti přístupu z internetu. Tím je odstraněna nutnost instalace aplikace na počítač a celkový přístup je optimálnější. Všichni uživatelé přistupují k téže aplikaci a pokud potřebujeme aplikaci upravit, stačí provést změny a nahrát je na webový server. Takto provedená změna se všem uživatelům projeví zároveň. Mezi online technologie spadají hlavně webové technologie, které jsou určeny především k prezentačním účelům.

3.5.1 Flash

Tato technologie má velký potenciál z hlediska prezentační formy. Nevýhodou je nutnost externího programu na straně uživatele, který je vyžadován pro spuštění flashové aplikace. Takový přístup mi přijde zbytečný. Další kritérium je, aby byla aplikace v případě potřeby snadno změnitelná či obohacena o další možnosti. Zde je opět problém jelikož se jedná o kompilovanou aplikaci a kdokoli by ji chtěl měnit, musí mít k dispozici zdrojové kódy. Jako poslední a asi nejdůležitější nevýhoda je potřeba placeného editoru, který by mohl dlouhodobý vývoj projektu zbytečně zatížit. Vzhledem k tomu jsem se rozhodl flashové aplikace ve své práci nevyužívat.

3.5.2 Java

Programovací jazyk Java má z hlediska podpory dobré možnosti. Existují pro něj editory, jako jsou NetBeans nebo Eclipse, které velmi usnadňují psaní zdrojového kódu a hledání případných chyb. Tento jazyk je vhodný pro naše účely, jelikož se aplikace v něm napsané dobře udržují a díky podpoře již zmíněných editorů i lehce rozšiřují. Nevýhoda této technologie je v problematickém spouštění, protože některé prohlížeče java applety nedokážou ani spustit. Někdy bývá problém s instalací pomůcek pro spouštění java appletů a běžný uživatel by nemohl potom aplikaci spustit. Byla by zde sice možnost desktopové aplikace ale v tomto případě bychom přišli o potenciál online přístupu.

3.5.3 Kombinace html, css, javascript

HTML bylo přísně založeno na SGML (Standard Generalized Mark-up Language), mezinárodně schválené metodě pro formování textu do strukturálních jednotek jako jsou odstavce, nadpisy, položky seznamu a tak dále. SGML může být implementováno na jakémkoliv počítači. Hlavní myšlenkou bylo, že jazyk bude nezávislý na klientu, který ve skutečnosti zobrazuje text na obrazovce. Používání párů značek (HTML tagů) je vzato přímo z SGML, které se chová totožně.

Během let 1993 a 1994 si mnoho webových prohlížečů přidávalo do HTML své vlastní prvky, čímž se tento jazyk stal špatně definovaným. V úsilí vytvořit pořádek v nastalém chaosu skupina lidí sjednotila všechny značky HTML, které byly široce užívány, a uspořádala je do pracovního dokumentu, který definoval šířku jazyka, který byl nazván HTML2.

V současné době je standardizována verze HTML 4.01, která je také poslední vývojovou verzí. Předpokládaným nástupcem HTML měl být jazyk XHTML založený na XML, ale s dalším vývojem HTML 5.0 je zatím XHTML paralelně vyvíjeno a stále se pracuje na verzi 2.0.

CSS slouží pro popis způsobu zobrazení stránek napsaných v jazycích HTML nebo XHTML. Jazyk byl navržen standardizační organizací W3C. Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Tyto informace byly čerpány z odkazu [3]

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní grafické prvky nebo tvořeny animace a efekty obrázků.

JavaScript je syntakticky podobný jazyku C, C++ a Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe.

Na rozdíl od jiných programovacích jazyků užívaných ve webových technologiích, JavaScript funguje na straně klienta, což znamená, že jeho interpret je zabudován v prohlížeči a nevyžaduje tudíž žádnou kompilaci. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

Jako další rozšíření této technologie přišel AJAX (Asynchronous JavaScript and XML) který umožňuje komunikaci se serverem, aniž by byla potřeba znovu přegenerovat stránku. Ke komunikaci se serverem se používají formáty XML, HTML nebo prostý text.

Jelikož je javascript interpretován na straně klienta, různé prohlížeče jej interpretují trochu jinak. Tím se stávají problémy, kdy kód fungující v jednom internetovém prohlížeči nefunguje v jiném internetovém prohlížeči. Proto byla vyvinuta snaha o odstranění těchto problémů a usnadnění práce s HTML dokumentem přes javascript. Jelikož většina operací, jež je třeba provádět se neustále opakuje, byly vytvořeny sady nástrojů, které tyto operace ulehčují. Tyto nástroje se nazývají „frameworky“. Jelikož jich existuje větší množství, budu se zde zabývat jen některými z nich.

Mootools poskytuje možnost rozšiřovat objekty základního modelu dokumentu HTML stránky. Jeho nevýhodou je zasahování mimo svůj vyhrazený prostor a tak vznikající možnost kolize s již existujícími javascriptovými kódy. Sice je poskytována celá řada rozšíření pro tento nástroj, ale pokud je jednou používán, je nebezpečné používat jiný způsob zápisu javascriptu či používat jiný nástroj.

Prototype se zaměřuje na jednoduchost vývoje dynamických webových aplikací. Představuje vlastní jednoduchý a unikátní systém používání tříd a přívětivou AJAXovou knihovnu. Má podobný přístup jako mootools a tedy i u tohoto nástroje může dojít ke kolizi s již existujícími javascriptovými kódy.

Jquery[4] slouží pro práci s AJAXem, manipulaci s HTML stránkou, práci s CSS a obsahuje podpůrné funkce například pro provádění animací. Ač je mladý, jeho potenciál je jednoznačný. Obrovskou výhodou je velikost, jelikož jádro je velmi malé. Je podporován všemi dnešními prohlížeči a stojí za ním velká komunita, která přispívá k jeho rozvoji a rychlému odhalení a odstranění případných chyb. Oproti předešlým nástrojům jquery pracuje ve svém prostoru a nehrozí kolize s ostatními kódy používanými ve stránkách. Má jednoduchý a robustní systém pro práci s událostmi prohlížeče. Taktéž i pro tento nástroj je vytvořena spousta rozšíření.

Po pečlivém zvážení jsem se přiklonil k použití nástroje jquery, jelikož jeho dynamický rozvoj, rychlost a syntaxe se mi líbil ze všech zmíněných nástrojů nejvíce.

3.6 Volba technologie pro uchování dat

Potřebujeme zvolit datovou vrstvu, kde budou data uložena tak, aby s nimi bylo možno jednoduše a efektivně pracovat. Nad těmito daty je třeba vytvořit architekturu, která obecně dokáže popsat většinu fotoaparátů a zároveň umožňovala jednoduchou změnou ovlivnit funkci výsledné aplikace. Tyto požadavky by mohly splňovat dvě technologie.

3.6.1 Databáze

Její výhodou je při dobrém návrhu lehká rozšiřitelnost záznamů a snadná manipulace s nimi. Jako velkou nevýhodu pro tento projekt považuji potřebu naprogramování rozhraní, které by usnadňovalo manipulaci s těmito daty. Neinformovaný člověk, který by chtěl jen lehce poupravit aplikaci (např. texty nebo přidat nějakou možnost do nastavení) by musel znát syntaxe jazyka pro komunikaci s databází a obecný způsob práce s databází. Proto by bylo nutné, vzhledem k dosud nspecifikovanému návrhu datové vrstvy aplikace, vytvořit nástroj, který by tuto funkcionalitu umožňoval. Tento nástroj by byl možná stejně složitý jako samotný simulátor, což by časově tento projekt zbytečně zažilo. Navíc by bylo také třeba rozhraní, které bude data interpretovat do podoby, jež bude srozumitelná aplikaci. Jelikož jsem si jako nástroj pro tvorbu simulátorů a trenážerů zvolil ajax/javascript, ideální by bylo data posílat ve formátu XML, který je v tomto případě ideální.

3.6.2 XML

eXtensible Markup Language – „rozšiřitelný značkovací jazyk“ je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C[3]. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem. Další možností je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML.

Jak již bylo zmíněno v předchozí části, použijeme tento značkovací jazyk jako datovou vrstvu pro naši aplikaci. Jelikož si můžeme libovolně nadefinovat strukturu dokumentu, kterou lze jednoduše v případě potřeby změnit, je tento jazyk ideální vzhledem k dynamickému vývoji aplikace.

3.7 Návrh dokumentu aplikace

Je třeba navrhnout patřičnou strukturu dokumentu popisující architekturu aplikace tak aby se obecně dala použít na většinu fotoaparátů a zároveň nebyla příliš složitá pro procházení či vyhledávání v dokumentu.

3.7.1 První návrh

První návrh zahrnoval myšlenku, že bych měl všechna data v jednom dokumentu a potřebné proměnné udržoval v cookies klienta prohlížeče. Začal jsem psát architekturu popisující menu nastavení fotoaparátu. Zde jsem ihned narazil na několik problémů. Jako hlavní problém byly chybějící informace o stavech fotoaparátu, které jsem se rozhodl držet na straně prohlížeče a ne ve svých dokumentech. Nebyl jsem tedy schopen jednoznačně určit stav, v kterém se fotoaparát aktuálně nachází a tím určit jaké možnosti jsou k dispozici.

Brzy jsem odhalil další nedostatky v tomto návrhu. Kromě chybějící možnosti jednoznačně definovat stav v jakém se aktuálně nacházím jsem narazil ještě na problém v základní myšlence, a to držet všechna data v jednom dokumentu. Jelikož se fotoaparát skládá z několika odlišných částí, není možné mít všechna data v jednom dokumentu, protože by se zde vyskytovaly různé architektury popisující části fotoaparátu a špatně by se s takovou datovou vrstvou pracovalo.

Ukázka prvního návrhu dokumentu:

```
<options>
  <option id="1" name="first" img="menu_play.gif" mod="1" >
    <item>
      <id>1</id>
      <name>Protect </name>
      <img>test</img>
    </item>
  </option>
</options>
```

3.7.2 Druhý návrh

Po předchozích problémech jsem tedy dokument rozdělil na několik logických částí dle funkcí fotoaparátu, které lze popsat odděleně.

3.7.2.1 Kontext

Jedná se o hlavní stav aplikace, v kterém se nachází. Tento stav určuje, jaké ovládací prvky jsou aktivní a jakou mají aktuálně funkci. Vzhledem k malému počtu ovládacích prvků (tlačítek) pro práci

s fotoaparátem mají některá tlačítka jinou funkci v modu prohlížení a jinou v modu focení. Taktéž při vypnutém fotoaparátu nemá většina tlačítek funkci žádnou.

3.7.2.2 Mód prohlížení nafocených fotografií

V této části potřebujeme udržovat pouze samotné nafocené fotografie, jelikož při jejich prohlížení není potřeba definovat žádné další záznamy týkající se ovládání. V dokumentu jsou uvedeny uzly „image“, jejichž obsahem je uzel „file“, který poskytuje fyzickou cestu k obrázku. V případě potřeby přidání, odebrání některých obrázků stačí přidat nebo odebrat uzel „image“. Pro potřebu změny aktuálních obrázků stačí pouze změnit cestu k souboru, která se uvádí od kořenového adresáře aplikace. Ukázka popisovaného dokumentu:

```
<content>
  <image>
    <file>./images/ulozene/obr_1.jpg</file>
  </image>
</content>
```

3.7.2.3 Menu nastavení fotoaparátu

Zde je již třeba držet více informací, jelikož nejsou všechny operace povoleny v každém stavu průchodu menu nastavení. To se skládá ze záložek a možností pod jednotlivými záložkami. Například, pokud uživatel je aktuálně u některé položky záložky, tlačítka „vlevo“ a „vpravo“ v tomto kontextu nemají žádnou funkci. Jsou aktivní tlačítka pouze pro pohyb směrem nahoru a dolů. Taktéž jsou zde výrazné grafické změny při procházení menu nastavení, které je třeba do tohoto dokumentu zahrnout. V dokumentu jsou dva hlavní prvky. Záložky reprezentované uzlem „fold“ a položky dané záložky které reprezentují uzly „item“ seskupené v uzlu „folder_items“. Jelikož u záložek byly rozpoznány 4 typy zobrazení, bylo potřeba je v dokumentu uvést. Tyto informace jsou u „fold“ pojmenovány jako „active_fold“ který obsahuje hodnotu pro zobrazení, pokud se pohybujeme aktuálně mezi záložkami a jsme na zvolené záložce. Další je „inactive_fold“ a tento uzel obsahuje hodnoty pro zobrazení ostatních záložek, pokud se aktuálně přepínáme mezi jednotlivými záložkami. Třetí uzel „chosen_fold“ obsahuje hodnotu pro zobrazení vybrané záložky v případě kdy, se pohybujeme mezi položkami dané záložky. Poslední uzel „disabled_fold“ obsahuje hodnotu pro zobrazení ostatních záložek v případě, kdy procházíme položky zvolené záložky. „Item“ je o něco jednodušší, jelikož nepotřebuje tolik informací a obsahuje pouze dva stavy. „Active_item“ obsahuje data pro zobrazení položky, která je v menu aktuálně vybrána. „Inactive_item“ obsahuje data pro všechny ostatní položky u dané záložky. Pro přidání další položky pod záložku stačí přidat nový uzel „item“, který bude mít stejnou architekturu jako ostatní uzly „item“. Pro odebrání položky stačí zrušit příslušný „item“. Pro přidání záložky platí stejná pravidla jako pro přidání položky. Na stejnou úroveň přidáme nový uzel „fold“ a nastavíme mu obsah již zmíněných uzlů „active_fold“, „inactive_fold“,

„chosen_fold“ a „disabled_fold“. Je zde ještě třeba přidat uzel „folder_items“ a do něj přidat nové uzly „item“. Tento uzel ovšem může být i prázdný. O formátu dat pro zobrazení se zmíním později, jelikož je společný pro více dokumentů a bylo by vhodné mu věnovat samostatnou část. Ukázka dokumentu:

```
<content>
  <fold id="1">
    <nazev>play menu</nazev>
    <active_fold>data</active_fold>
    <inactive_fold> data </inactive_fold>
    <chosen_fold> data </chosen_fold>
    <disabled_fold> data </disabled_fold>
    <manual>nastaveni/fold_play.html</manual>
    <folder_items>
      <item id="10">
        <active_item> data </active_item>
        <inactive_item> data </inactive_item>
        <manual>nastaveni/protect.html</manual>
      </item>
    </folder_items>
  </fold>
</content>
```

3.7.2.4 Režimy focení

Změna režimu focení ovlivňuje především možnosti, které lze u fotografie nastavit. Například u automatického režimu máme pole působnosti pro ovlivnění výsledné fotografie výrazně omezené, na druhé straně v manuálním režimu můžeme nastavit téměř vše. Tyto nastavení se dělí do dvou částí. Obecnou a funkce. V obecné části lze u fotografie nastavit možnost focení na makro, možnost s bleskem nebo nastavit dobu expozice. To vše bylo třeba do dokumentu zahrnout. Jako základní údaj u každého z režimů musí být uveden uzel „name“ kde je obsažen název režimu. Dále je pak zde uveden název obrázku, který se při změně režimu zobrazí na displeji. Ten je uveden v „image“ a uvádí se bez cesty, jelikož obrázky se budou měnit pouze na základě změny adresáře. „Ico“ obsahuje cestu k ikoně, která reprezentuje tento režim. Jako další informace jsou zde uvedeny elementy pro funkce ovlivňující vzhled fotky v „obecné části“. „Flashlight“ obsahuje informace k prezentaci fotoaparátu při focení s bleskem. Jeho synovský uzel „ico“ zobrazuje na displeji ikonu blesku a „image“ obsahuje název fotky, jež reprezentuje daný efekt na fotografii. Uzlů „flashlight“ může zde být více, ovšem nejméně dva, jelikož tato hodnota může být zapnutá nebo vypnutá a musí tedy být obsažena v tomto dokumentu. Dalším uzlem je „macro“ který určuje zobrazení efektu jak by fotka

vypadala při focení s makro režimem. Struktura tohoto uzlu je shodná s uzlem „flashlight“. Rozdíl je v počtu možností, jelikož blesk může mít i více než dvě volby (s bleskem, bez blesku, s redukcí červených očí, ...), voleb makro mohou být pouze dvě a to zda jej používáme či nikoliv. Poslední uzel „expoze“ má totožnou strukturu s uzly „flashlight“ a „macro“. V simulaci slouží pro vysvětlení funkce doby expoze. Ukázka dokumentu:

```
<content>
  <mode>
    <name>manual</name>
    <status>ok</status>
    <manual>rezimy/manual.html</manual>
    <image>on.jpg</image>
    <ico>data</ico>
    <flashlight>
      <ico>data</ico>
      <image>on.jpg</image>
      <manual>rezimy/manual.html</manual>
    </flashlight>
    <macro>
      <ico>data</ico>
      <image>on.jpg</image>
      <manual>rezimy/manual.html</manual>
    </macro>
    <expoze>
      <ico>data</ico>
      <image>on.jpg</image>
      <manual>rezimy/expoze.html</manual>
    </expoze>
  </mode>
</content>
```

3.7.2.5 Funkce fotoaparátu

Pod tímto pojmem je zahrnuta druhá část nastavení fotografie, která je ovlivněna zvoleným režimem fotoaparátu. Zde se nacházejí doplňková nastavení, jako jsou například výstupní kvalita fotografie, rychlost spouště a další. Při bližší analýze této části se dostaneme k závěru, který nám usnadní práci. Každé z těchto nastavení představuje pouze sadu předdefinovaných parametrů, z kterých si můžeme jeden vybrat. Vycházejí vždy ze stejné architektury a liší se pouze v zobrazených parametrech. Z této analýzy odvodíme architekturu xml dokumentu. Také do tohoto dokumentu zapíšeme informace o

aktuálně zvolených nastaveních. Jelikož informace v tomto dokumentu jsou přímo spojeny s režimy fotoaparátu, potřebujeme někde uvést, která nastavení jsou dostupná pro daný režim. Pro tento účel si vytvoříme speciální uzel, ve kterém budeme uvádět přístupnost nastavení v jednotlivých režimech. Obsahuje uzly „value“ které reprezentují jednotlivé režimy. Obsahem uzlu jsou záznamy pro jednotlivá nastavení. Ty mohou nabývat třech různých hodnot. Hodnota „visible_only“ říká, že nastavení je zobrazeno, ale nelze změnit jeho parametry. Hodnota „disable“ zajistí skrytí tohoto nastavení, aby se nezobrazovalo. Poslední hodnotou je „0“ která znamená zobrazení nastavení a možnost změny jeho parametrů. Tento uzel umístíme na začátek dokumentu, kde jej lehce identifikujeme. Každý uzel reprezentující některé z nastavení obsahuje povinné „name“, „image“ a „visible_only“. „Image“ obsahuje implicitní obrázek, který se zobrazí pro tuto možnost nastavení. „Visible_only“ obsahuje cestu k obrázku, jež se zobrazí, pokud nastavení nebude možno v aktuálním režimu změnit. Uzel „name“ obsahuje oficiální název tohoto nastavení. Dále jsou zde uzly „value“ které reprezentují jednotlivé parametry (hodnoty) těchto nastavení. Parametr má uzel „name“ obsahující název tohoto parametru. Uzel „active_ico“ obsahuje data pro zobrazení parametru, když je vybrán a uzel „inactive_ico“ data pro zobrazení stavu kdy vybrán není. V uzlu „image“ je název obrázku, na kterém demonstrujeme efekt, jež tento parametr vytváří. Posledním údajem nastavení je uzel „action“, který popisuje povolení operací pro jednotlivá tlačítka ovládání fotoaparátu. Pro ukázkou uvádím část kódu speciální položky, jež definuje dostupnost nastavení pro jednotlivé režimy:

```
<item>
  <value>
    <accessible_setting>
      <enable>disable</enable>
    </accessible_setting>
    <accessible_setting>
      <enable>0</enable>
    </accessible_setting>
  </value>
</item>
```

Na další ukázce je zobrazena část kódu popisující nastavení s jeho parametry:

```
<item>
  <name>flash_output</name>
  <image>./images/foto/2.jpg</image>
  <visible_only>data</visible_only>
  <value>
    <name>something</name>
    <image>./images/foto/2.jpg</image>
    <active_ico>data</active_ico>
    <inactive_ico>data</inactive_ico>
  </value>
</item>
```

3.8 Implementace

Po definování finální datové struktury xml dokumentů přejdeme k části implementace jednotlivých funkcí simulátoru. Protože hlavní funkcionality fotoaparátu nejsou na sobě povětšinou závislé, nebo se dají rozdělit do logicky oddělitelných celků, můžeme se těmito částem věnovat odděleně a po jejich dokončení je sloučit do výsledné aplikace. V návrhu aplikace byl zmíněn nástroj pro tvorbu simulátorů. To lze uskutečnit pomocí námi definovaných dokumentů. Musíme naprogramovat také aplikační kód, který bude s těmito daty rozumně pracovat. Následně provázeme datovou vrstvu s grafickou, jak již bylo zmíněno v návrhu tohoto nástroje.

3.8.1 Modulární vývoj

Jak již bylo řečeno, rozdělíme aplikaci na jednotlivé moduly a každý vypracujeme zvlášť. I když jsou všechny moduly na sobě nezávislé, mají některé společné prvky, které je třeba definovat zvlášť. Jedním z těchto prvků je ovládání aplikace, které je společné pro všechny moduly. Z toho důvodu definujeme konstanty pro tyto ovládací prvky a ve všech modulech je budeme používat. Tímto přístupem si zajistíme zpětnou funkčnost po sloučení všech modulů, protože budou používat jednotný systém. Dalším z těchto společných prvků jsou hodnoty reprezentující stav fotoaparátu. Všechny tyto proměnné vložíme do souboru, který bude obsahovat globální údaje potřebné pro běh aplikace. Po tomto kroku již můžeme přistoupit k samotným modulům aplikace.

3.8.1.1 Modul menu nastavení fotoaparátu

Zde se budeme zabývat částí nastavení fotoaparátu. Procházení menu nastavení se dá popsat jako průchod přes dvojrozměrné pole. První index znázorňuje záložky a druhý jednotlivá nastavení. Pokud

se podíváme na dokument popisující menu nastavení, je zde vidět stejný princip. Dále je třeba definovat pravidla pro průchod nastavením. Pokud se pohybuji v záložkách, mám možnost pohybu do všech stran. Pokud se pohybuji v položkách záložky, mám možnost se pohybovat pouze nahoru nebo dolů, protože funkcionalita pro tato tlačítka není implementována. Z toho vyplývá nutnost definovat si dvě proměnné. Jedna reprezentuje aktuálně zvolenou záložku a druhá aktuálně zvolenou položku. Nyní zde musí platit pravidlo, když jsem na nějaké položce, nelze se pohybovat mezi záložkami. Proto je třeba si u proměnné pro pohyb v položkách definovat speciální hodnotu. Ta bude nastavena v případě, kdy není zvolena žádná položka, a tedy se pohybujeme mezi záložkami.

3.8.1.2 Modul nastavení funkcí pro fotografování

Pro tento modul je používán pokročilý návrh datové struktury. Obsahuje nejen data, ale i definici povolených akcí na těchto datech. Jak jsme se již zmínili, v části zabývající se dokumentem nastavení funkcí, je zde speciální položka obsahující hodnoty dle kterých se nastavení zobrazí či nikoliv. Pokud je v poli obsažena hodnota „disable“, nastavení se ignoruje. Pokud je zde „visible_only“, položka se zobrazí, ale nelze jí měnit parametry. V posledním případě je toto nastavení zobrazeno a do pole se při průchodu nastavením ukládá hodnota aktuálně zvoleného parametru. Při stisku tlačítka se zjišťuje, zda má pro toto tlačítko aktuálně zvolené nastavení definovanou akci a pokud ano, tak je tato akce provedena.

Obsah displeje je při změně hodnoty nastavení vždy generován znovu. Zkoušel jsem obsah displeje vygenerovat pouze jednou a pak s ním pracovat. V brzké době jsem narazil na problém nekonzistence mezi stavem na datové vrstvě a stavem na displeji. Proto je jednodušší obsah generovat vždy znovu. Tento přístup je používán i u modulů menu nastavení a procházení nafocených fotografií.

3.8.1.3 Modul změny režimu

Změna režimu funguje podobně jako předchozí dva moduly. Při změně režimu fotoaparátu se do globálně držených proměnných uloží index režimu, který je využíván také v předchozím modulu. Následně se na displeji zobrazí fotka reprezentující aktuální režim. Modul současně obsahuje i data pro obecnou část nastavení focení jako blesk či makro. Při stisknutí příslušných tlačítek na ovládání fotoaparátu se zobrazí na displeji ikonky pro znázornění zapnutí této možnosti a zároveň se zobrazí obrázek, kde bude patrný efekt, který by s fotkou tato možnost provedla.

3.8.1.4 Modul procházení nafocených fotografií

Tento modul demonstruje způsob procházení nafocenými fotografiemi a možnosti jejich prohlížení. Při přibližování se na fotografii uplatňuje zoom a je možné si prohlédnout její detail. Ten je omezen určitou hodnotou. Při oddalování se nejprve zobrazí 9 fotografií, po kterých se můžeme pohybovat. Při druhém oddálení je možné listovat přes nafocené fotografie po 9 fotografiích. K tomu musíme

definovat konstanty udávající maximální kladný a záporný zoom v módu prohlížení. Dále potřebujeme proměnnou udávající aktuální hodnotu zoomu a proměnné pro souřadnice umístění displeje na přiblížené fotografii. Následně při přiblížení fotografie přepočítáme zvětšení a zaostříme na střed aktuálního zobrazení. Pohybem po přiblížené fotografii měníme pouze souřadnice, a pokud narazíme na okraj fotografie, nedovolíme další pohyb v tomto směru.

3.8.1.5 Modul zoom

Zoom nemá žádnou vlastní datovou vrstvu, protože pracuje jen s aktuálně zobrazenou fotografií na displeji. Pro svoje správné fungování potřebuje znát maximální velikost podporovaného přiblížení. Díky Jquery a jejímu nástroji pro tvorbu animací lze vytvořit snadno a rychle zoom, který bude vypadat jako skutečný. Při vývoji tohoto modulu nastal drobný problém ve chvíli, kdy jsme stiskli tlačítko pro změnu zoomu v průběhu přibližování. Nelze provádět další přiblížení či oddálení dokud se nedokončila aktuální změna přiblížení. Zde pomohl selektor „animated“, kterým zjišťujeme, zda změna přiblížení právě probíhá nebo lze začít novou.

3.8.1.6 Modul změny grafického téma

Změna grafického režimu nemá žádnou spojitost se samotnou simulací. Dodává ovšem celé práci uživatelskou přívětivost. Díky tomuto modulu si uživatel může zvolit, na jakých fotografiích si bude chtít prohlédnout efekty, jež fotoaparát poskytuje. Sady fotografií reprezentující zvolené nastavení aparátu mají totožně pojmenovány fotografie se stejným efektem. Tyto sady jsou rozděleny do adresářů. Modul při kliknutí na motiv změni adresář, z něhož se mají fotografie používat.

3.8.1.7 Modul jazykové lokalizace

Tento modul umožňuje vícejazyčnost aplikace. Jeho xml dokument obsahuje název jazyka, ikonu a cestu k manuálům, kterými se budeme zabývat později. Dále jsou zde uvedeny všechny texty, jako jsou popisy tlačítek či kontextová nápověda pro tlačítka atd. Při změně jazyka se ihned načtou všechny texty pro aktuálně zvolený jazyk.

3.8.1.8 Modul ovládání aplikace

Zde nadefinujeme operace a kontexty pro jednotlivá tlačítka, s výjimkou těch co jsou na ovládacím panelu fotoaparátu. Jelikož chování tlačítek je ovlivněno kontextem simulátoru, budou na každém tlačítku podobné funkce. Ty vytvoří stavy aplikace, kam se při sloučení modulů nakopírují volání jednotlivých funkcí, jež se mají v daném stavu provádět.

3.8.2 Práce s datovou vrstvou

Při práci s xml dokumenty jsem narazil na problém zpracování dat ve formátu xml a jejich efektivního čtení a zápisu zpět do dokumentu. Javascript přímo nabízí funkce pro práci s xml dokumentem. Jejich nevýhodou je individuální způsob zpracování na prohlížeči, který není kompatibilní. Bylo by nutné napsat každému prohlížeči vlastní kód pro zpracování těchto dokumentů. Další variantou bylo zpracování dokumentů přes JQuery.

Jelikož xml dokument má stejný způsob zápisu jako html dokument, lze na něj aplikovat přístup, jež JQuery nabízí. Tím jsou selektory používané v kaskádových stylech. Pomocí nich lze procházet strukturu dokumentu, vyhledávat v něm a pracovat s takto nalezenými prvky. Více se o tomto způsobu dá dočíst v odkazech [3] a [4]. Se selektory se pracovalo lépe než s javascriptovými funkcemi pro práci s xml dokumenty, ovšem i zde chyběly některé možnosti. Ve většině případů bylo potřeba zacházet s dokumentem jako s polem, mít možnost jej indexovat a tímto způsobem ho i procházet. Proto jsem začal hledat způsob jak převést xml dokument do javascriptového pole. Zde jsem objevil technologii, která splňuje moje požadavky.

3.8.2.1 JSON

Jedná se o univerzální datové struktury a v podstatě všechny moderní programovací jazyky je v nějaké formě podporují. Je tedy logické, aby byl na nich založen i jazykově nezávislý výměnný formát.

JSON je založen na dvou strukturách:

- Kolekce párů název/hodnota. Ta bývá v rozličných jazycích realizována jako *objekt*, *záznam*, *struktura*, *hash tabulka*, *klíčový seznam* nebo *asociativní pole*.
- Tříděný seznam hodnot. Ten je ve většině jazyků realizován jako *pole*, *vektor*, *seznam* nebo *posloupnost*.

Tyto informace jsem čerpal z internetového zdroje[5].

Pomocí volně dostupného parseru z xml do JSON struktury se vytvoří javascriptové asociativní pole, které již umožňuje procházet data přes indexy.

3.8.3 Sloučení modulů

U sloučení modulů musíme dbát na správné pojmenování proměnných. Sdílená data již máme připravena v samostatném souboru a data pro jednotlivé moduly jsou v příslušných souborech. Díky oddělenému vývoji by se po sloučení mohlo stát, že si moduly budou navzájem přepisovat stejně pojmenované proměnné. Proto všechny tyto proměnné a konstanty umístíme do jednoho dokumentu, kde budou přehledně zapsané. Další problém při sloučení modulů se projevuje ve chvíli, kdy

vkládáme zdrojové soubory do výsledné aplikace. Ty musejí být uvedeny ve správném pořadí, protože jinak by se mohla nějaká funkcionality snažit volat funkci, která bude načtena až později.

3.9 Dodržení zamýšleného postupu

Původní návrh realizace projektu byl jiný, než jsem nakonec udělal. Chtěl jsem vytvořit nástroj pro tvorbu simulátorů, v kterém by se potom implementoval simulátor Canon A 70. Z tohoto postupu se zachovala pouze část původní myšlenky. Vytvořil jsem simulátor, který splňuje požadavek na snadnou rozšiřitelnost o nové funkce. Tím lze na tomto simulátoru demonstrovat i podobný typ fotoaparátu. Co se týká nástroje na tvorbu simulátorů jako takového, vznikla pouze konvence pro styl zápisu dokumentu. Některé kódy, jako je modul zoom, by šlo jednoduše znovu použít u jiných simulátorů. U jiných kódů, jako je například modul menu nastavení, by bylo potřeba udělat podstatně více změn. I když se původní záměr nepodařil, je toto krok správným směrem.

4 Manuál

V této části se budeme zabývat tvorbou manuálu a návodu pro fotoaparát Canon A70. Musíme si uvědomit, jaké jsou požadavky na manuál a komu bude sloužit a jaký je jeho účel.

4.1 Požadavky

Tento manuál by měl přiblížit možnosti fotoaparátu Canon A70 z hlediska jeho prezentace. Manuály dodávané prodejcem jsou příliš technické a základní informace o fotoaparátu se v nich jen těžko hledají. Jelikož má sloužit hlavně starším lidem mělo by se klást důraz na jednoduchost a přehlednost.

4.2 Návrh

Ve srovnání s ostatními bakalářskými pracemi jsem pojal manuál trochu jinak. Všichni ho měli jako externí dokument, který měl zobrazovat to samé co simulace. Tím jsem přišel s nápadem spojit simulátor a manuál v jedno. Při procházení aplikace se v každém okamžiku bude zobrazovat manuál s vysvětlením aktuálního stavu nebo s informacemi, co je možné v tomto stavu provádět. Takto zobrazený manuál má mnohem větší využití, protože je používán přímo v praxi a uživatel má možnost si funkce, o kterých čte v manuálu, ihned vyzkoušet.

4.3 Provedení

Manuál je naimplementován jako textové soubory, jejichž obsah je ve formátu HTML. Díky tomuto stylu zápisu lze manuály lehce graficky upravit, vkládat do nich obrázky, seznamy a další prvky, jež HTML umožňuje. Taktéž je možnost vkládat odkazy na externí zdroje, protože místo složitého opisování některých informací raději uživatele odkážeme na místo, kde si tyto informace může přečíst. Nesporná výhoda je v možnosti upravit si text manuálu dle vlastní potřeby a uvážení, protože jsou umístěny mimo jádro aplikace a jsou psány v obecně známé technologii.

Se simulátorem jsou manuály provázány přes uzel „manual“ který je uváděn u každé záložky a nastavení. Pokud manuál pro některou položku neexistuje, vypíše se o tom patřičná zpráva.

Manuál je taktéž provázán s jazykovou lokalizací. Ta je realizována podobně jako obrázky pro změnu motivu aplikace. U souborů musí být zachován název, protože rozdíl mezi českou a anglickou verzí je v adresáři, z kterého se soubory čtou. Název adresáře je uveden v xml dokumentu jazykových lokalizací.

5 Ostatní bakalářské práce na téma simulátor a trenžér digitálních fotoaparátů

V této kapitole se budeme zabývat ostatními pracemi na téma simulátor a trenžér digitálního fotoaparátu. Protože před mojí bakalářskou prací bylo již jedenáct stejných témat úspěšně obhájeno, je na místě jejich vzájemné srovnání. Postupně budeme porovnávat práce, zdůrazníme největší rozdíly, ale i společné principy. Dále se ve stručnosti zde dozvíme, v čem se liší moje bakalářská práce.

5.1 Manuály

Součástí každého zadání byla i tvorba manuálu pro zadaný fotoaparát. Zde většina studentů dodala podobné návody ve formě pdf nebo prezentace powerpoint. Tuto podobu studenti zvolili pro její vhodné využití při výuce a to z důvodu přání pana doc. Kunovského tyto materiály použít pro lekce na univerzitě třetího věku.

Protože klasická forma technického manuálu není vhodná ani jako učební materiál, ani pro starší lidi bez technického vzdělání, tak studenti dbali pravidel přístupnosti. Většina manuálů se liší pouze rozsahem, které materiály pokrývají.

Nicméně i zde se našel odlišný přístup a to vytvoření návodu ve flashi, avšak takovéto manuály se mi subjektivně jevily jako méně přehledné. Další zajímavý přístup jsou návody ve formě videí, často v kombinaci se simulací.

5.2 Simulátory

Pokud budeme srovnávat simulátory, tak si všimneme větší pestrosti nabízených řešení. Navíc je velký rozdíl mezi pracemi z letošního a minulého roku, kdy letošní práce mnohdy dosahují profesionální úrovně i z grafického hlediska. Minulý rok byli dosažené výsledky o něco horší, ačkoli je to způsobeno i tím, že letošní práce na ty předchozí navazovaly.

Jedno však mají téměř všechny simulace, až na jednu, společné. Jsou zaměřeny na využití internetových technologií a snadnou přístupnost na www. Jediná bakalářská práce toto pravidlo porušuje, je tvořena jako desktopová aplikace.

5.2.1 Zaměření

V tomto bodě se podíváme, na co se autoři při tvorbě zaměřili. Jsou zde práce, které se zaměřují na grafickou stránku věci, téměř až na profesionální úrovni. Tyto práce většinou nezahrnují simulaci fotoaparátu příliš komplexní. Ostatně v některých případech by to již bylo nepřehledné. Na druhé straně jsou zde práce, které se zaměřují na přesnost a bohatost poskytnutých informací o simulovaném fotoaparátu. Tyto práce se pak dají využít při výuce.

Dalo by se tedy říci, že jsou tu dva principy, jeden je využití simulace v komerční sféře a druhý jako podpůrný materiál pro výuku. Málokterý trenažér splňuje oba tyto principy. Další věcí, které si můžeme všimnout je celkový dojem z aplikace. Ty simulace, které toho implementují méně, působí ucelenějším dojmem a zdají se být propracovanější. I když se i v tomto pravidle se najde výjimka jej potvrzující a to práce Lubora Páče, která je komplexní a zároveň ucelená.

5.2.2 Společné myšlenky a postupy

Téměř všechny simulace mají společné cíle. Jedním z těchto cílů je i snadná dostupnost na internetu. Z těchto důvodů velká část studentů zvolila stejnou technologii. Tímto nástrojem je jazyk Flash, případně jeho volně dostupná varianta Flex. Oba nástroje jsou určeny pro tvorbu grafiky.

5.2.3 Různé nápady

Ještě jednou zde zmíníme Lubora Páče, který do simulace přinesl velmi originální myšlenku a to zapojení webkamery. Povolíme-li jeho aplikace k webkameře přistupovat, pak místo fotografie se na pozadí displeje zobrazí obraz z kamery.

Dále je v některých pracích vidět jak se projevují jednotlivá nastavení na vzhledu výsledné fotografie.

5.2.4 Odlišné práce

Odlišnou práci od všech ostatních obhájil Viktor Michálek, který se jako jediný zaměřil na desktopovou aplikaci se zcela 3d modelem. Toto pojetí má své výhody i nevýhody. Výhodou je to, že uživatel získá přesnější představu o fotoaparátu. A mohlo by přinést i další výhody, jako třeba menší nároky na počítač než častou používaný flash. Nicméně zde byla použita nevhodná technologie a simulace je velmi náročná.

5.3 Moje bakalářská práce

Svoji bakalářskou práci jsem pojal oproti ostatním jinak hned v několika bodech.

5.3.1 Volba implementačního nástroje

Pro tvorbu své bakalářské práce jsem použil tyto technologie: Javascript, HTML, CSS, XML. To je první věc kterou je má simulace odlišná od ostatních.

5.3.2 Manuál a simulace

Ve své práci jsem spojil manuál a simulace do jednoho celku a to tak, aby uživatel přímo v simulaci viděl co daná akce udělá a jaký je aktuální stav. V manuálu je nápověda pro jednotlivé režimy focení, položky menu, případně detailní nastavení fotoaparátu.

Je pravda, že v některých simulacích bylo již takové spojení téměř implementováno, nicméně s jednou odlišností, uživatel se musí přepnout do jiného okna simulace.

5.3.3 Kontextová nápověda pro tlačítka

Další věcí, kterou jsem jinde neviděl a která se mi zdá být užitečná, je kontextová nápověda pro ovládací tlačítka fotoaparátu. To uživateli usnadní orientaci, protože hned vidí, co bude následovat, pokud zmáčkne vybrané tlačítko.

5.4 Zhodnocení a další pokračování tématu

Ve výše uvedených bakalářských pracích bylo zmíněno několik zajímavých aplikací a je zajímavé vidět, jak si s nimi jednotliví studenti poradili. I když velmi často zvolili po konzultacích s panem doc. Kunovským podobné řešení, tak výsledky jsou většinou velmi odlišné. Osobně si myslím, že takto zadaná bakalářská práce byla pro studenty přínosná.

Jako pokračování tohoto tématu vidím možnost vytvoření nástroje na tvorbu těchto simulací. Moje práce, ačkoli přináší základy, tak tímto nástrojem zatím není. Především zde není automatizace psaní XML, některé moduly jsou spojeny poněkud těžkopádně.

6 Možná rozšíření

Při závěrečných úpravách jsem narazil na některé možnosti, které by mohly projekt obohatit. Bohužel se již na ně nedostalo. Proto bych je zde rád zmínil, kdyby někdo chtěl pokračovat v mém úsilí.

6.1 Zrychlení aplikace

Protože aplikace obsahuje velké množství obrázků, při prvním načtení může být pomalá. To způsobují obrázky, které se načítají v průběhu simulace, jak jsou zrovna potřeba. Tento problém by vyřešilo načtení všech obrázků ještě před začátkem používání aplikace.

6.2 Znalostní báze

Pro projekty tohoto typu by mělo cenu založit nějakou znalostní bázi, kde by se uchovávaly informace spojené s digitálními fotografií. Do této báze by se dalo lehce odkazovat z manuálu. Ve výsledku by se potom sjednotila data, která budou poskytována uživatelům.

7 Závěr

Výsledkem této práce je simulátor fotoaparátu Canon A70. Vytvořenou aplikaci hodnotím kladně, protože je zde několik nových myšlenek. Měl jsem možnost nahlédnout na ostatní bakalářské práce na toto téma a některými nápady se inspirovat. Jako můj hlavní přínos pro projekt vidím v částečně implementovaném nástroji pro tvorbu simulátorů. Tento systém umožní drobnou úpravou simulovat fotoaparát podobné řady a nebude třeba k tomu účelu psát nový simulátor. Jako novou myšlenku považuji spojení manuálu a simulátoru. Z hlediska výuky je lepší pokud uživatel si rovnou zkouší nově nabyté zkušenosti.

Z hlediska dalšího vývoje projektu již nemá smysl tvořit další simulátory ale zaměřit se na tvorbu nástroje pro tento účel. Moje řešení je o krok blíže k tomuto stavu. Výsledný nástroj pro tvorbu simulátorů by měl umět nejen simulovat zvolený fotoaparát, ale měl by umožňovat tvorbu jednoduchých animovaných scénářů jak s tímto fotoaparátem zacházet.

Přínos bude jednoznačně pro studenty univerzity třetího věku, pro které byl tento simulátor tvořen. Při srovnání s ostatními bakalářskými věřím, že tento přístup bude do budoucna více využíván.

Literatura

[1] Kolektiv autorů. Webové stránky fotografování. <http://www.fotografovani.cz>.

[2] Kolektiv autorů. Webové stránky paladix. <http://www.paladix.cz>.

[3] World wide web consortium. <http://www.w3.org>

[4] Resig, J.: *jQuery*. 2009, <http://jquery.com>

[5] Kolektiv autorů. Webové stránky json <http://www.json.org>

Seznam příloh

Příloha 1. Simulátor a trenažér fotoaparátu Canon A70