

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENERÁTOR KOMIXŮ - MASHUP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ KRUTIŠ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GENERÁTOR KOMIXŮ - MASHUP

COMIC STRIP GENERATOR - MASHUP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ KRUTIŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. **RUDOLF KAJAN**

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá tvorbou webové aplikace, ve které si uživatel může vytvořit a uložit komiks. Jako zdroje obrázků pro komiks jsou použity služby na internetu, které poskytují zdroje obrázků přes rozhraní těchto služeb. Jedná se o mashup aplikaci. Práce se zabývá přiblížením použitých technologií, návrhem samotné webové aplikace a ukazuje řešení implementace některých částí aplikace.

Abstract

This Bachelor thesis describes a creation of a web application, in which can user create and save comics. As a sources of images are used web services, which provide images via their API. This application is mashup. The work describes applied technologies, design of web application and shows the implementation of some parts of the application.

Klíčová slova

Komiks, mashup, obrázky, bubliny, Silverlight, API, Flickr, Picasa

Keywords

Comics, mashup, pictures, bubbles, Silverlight, API, Flickr, Picasa

Citace

Krutiš Tomáš: Generátor komixů - Mashup, bakalářská práce, Brno, FIT VUT v Brně, 2011

Generátor komixů - Mashup

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Rudolfa Kajana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Krutiš
18. 5. 2011

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Rudolfovi Kajanovi za odbornou pomoc a rady.

© Tomáš Krutiš, 2011

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
1 Úvod.....	3
2 Komiks, Web 2.0 a Mashup.....	4
2.1 Komiks.....	4
2.2 Web 2.0.....	5
2.3 Mashup	6
2.3.1 REST.....	7
3 Použité technologie	8
3.1 Silverlight	8
3.1.1 Popis technologie Silverlight	9
3.1.2 Vývojové nástroje	9
3.1.3 Historie technologie Silverlight	10
3.2 Pixel shader – HLSL.....	10
3.3 PHP.....	11
4 Návrh aplikace	12
4.1 Komiks a jeho ovládání	12
4.1.1 Posuvníky	13
4.1.2 Pozice obrázku v panelu	14
4.2 Výběr obrázku a jeho editace.....	14
4.2.1 Zdroje obrázků.....	15
4.2.2 Bubliny	19
4.2.3 Kresba do obrázku	20
4.2.4 Efekty.....	21
5 Implementace.....	24
5.1 Třída Komix.....	24
5.2 Třída KomixImage.....	25
5.3 Třída PoziceObrázku	26
5.4 Třída Bublina.....	27
5.5 Třída FlickrPhoto.....	27
5.6 Efekty.....	28
5.7 Picasa Wrapper.....	30
5.7.1 Přístup k webovým albům Picasa	30
5.7.2 Vyhledávání	31
5.7.3 Přihlášení do webových alb Picasa.....	31

5.7.4	Alba přihlášeného uživatele	32
5.7.5	Výběr alba.....	32
6	Závěr	34
6.1	Možná budoucí rozšíření	34

1 Úvod

Internet prošel za poslední roky velkými změnami. Již to nejsou pouze statické stránky poskytující informace. Technologie umožňují vytvářet interaktivní webové aplikace, které dříve bylo možné vytvářet pouze jako aplikace nainstalované na počítači. Technologie, jako je Silverlight poskytují prostředky pro vytváření interaktivních webových aplikací běžících v internetovém prohlížeči s využitím zásuvného modulu.

Služby provozované na internetu poskytují rozhraní pro přístup k jejich datům. Např. služby, na které si uživatel ukládá své fotografie nebo služby poskytující mapové podklady. Tím získáme obrovský zdroj dat, a pokud k dané službě je dostupné rozhraní tzv. API, můžeme tyto data využít. Takže můžeme vytvářet tzv. mashup webové aplikace, které kombinací dat z těchto zdrojů vytvoří novou službu. Bez zdrojů dat z jiných služeb by takové služby vznikali daleko hůře.

Tato bakalářská práce se zabývá vytvořením webové aplikace, která by dokázala vytvořit komiks právě kombinací obrázků a fotografií z různých zdrojů. Práce se zabývá vytvořením schématu komiksu, získáváním obrázků do komiksu z pevného disku počítače nebo ze služeb na internetu a dovolí uživateli upravovat získané obrázky. Po dokončení kompozice obrázků do schématu komiksu se bude moci hotový produkt uložit na pevný disk počítače. Jsou zde také popsány použité technologie a postupy při vývoji webové aplikace pro generování komiksů.

2 Komiks, Web 2.0 a Mashup

Tato práce se zabývá tvorbou komiksů, které kombinují obrázky z různých zdrojů. Není tedy od věci vysvětlit, co vlastně tyto termíny znamenají a co si pod nimi má čtenář představit.

2.1 Komiks

Komiks je umělecké médium, ve kterém jsou obsaženy kresby nebo jiná vyobrazení. Jsou doplněná textem, jenž vysvětluje nebo popisuje, co se v komiksu odehrává. Ve finále tedy většinou vypráví nějaký příběh.

Jednotlivým obrázkům, které zachycují konkrétní momenty příběhu, se říká panely. Podmínkou komiksu je, že jednotlivé panely musejí být rozloženy tak, aby se nacházely vedle sebe v jednom konečném obrázku. Tím se odlišuje komiks od animovaného filmu.

Komiksy jsou charakteristické používáním „bublin“. Tyto prvky slouží k vyjadřování projevů a myšlenek postav komiksu nebo jako texty pronesené vypravěčem příběhu. Bubliny a texty jsou součástí jednotlivých panelů.

Krátké komiksy do šesti panelů bývají označovány jako „strip“. Jsou většinou vkládány do spodní části stránky v novinách. Anglické slovo „comics“ vzniklo právě kombinací slov „comic“ a „strip“, což znamená něco jako vtipný nebo zábavný proužek. Do češtiny bylo toto slovo přejato jako „komiks“ nebo „komix“.

I když se komiks začal prosazovat ve 20. století, jeho prapředky můžeme najít již ve starověkém Egyptě jako nástěnné malby, na nichž je zobrazena nějaká část příběhu v jednotlivých zobrazeních za sebou. K většímu rozšíření došlo až s příchodem knihtisku, kdy začali vznikat první kreslené série.

První novodobý komiks vyšel 16. 2. 1896 v časopise New York World, nesl název Žluté dítě a jeho autorem je Richard Outcault. Den kdy vyšel tento krátký komiks je také označován jako „Den zrození komiksu“. [1]

Komiksy jsou populární hlavně ve světě, u nás je problém s předsudkem, že komiks je něco, co je určené pro děti. Je to z toho důvodu, že v minulosti nevycházelo mnoho komiksů pro dospělé. Hlavně v Americe je velká komiksová komunita, která se o tento druh zábavy hodně zajímá. Komiksové předlohy slouží v hojné míře jako předloha pro řadu filmů a seriálů, které z převážné většiny pochází ze Spojených států amerických (např. Iron Man, série X-Men, Transformers atd.). Mezi nevěhlasnější vydavatelství komiksů patří např. DC Comics a Marvel Comics.

Existují také tzv. „webové komiksy“, které vychází jednou denně nebo týdně pouze na webu. Jedná se o krátké komiksy (2-5 panelů), které zobrazují jednoduché, vtipné příběhy (např. Garfield, ExtraLife, ...).

2.2 Web 2.0

Termín Web 2.0 nebyl zatím nikdo schopen definovat standardním způsobem, je to spíše soubor principů, praktik a vlastností, které dávají dohromady význam pojmu Web 2.0.

Web 2.0 byl evolučním krokem, který musel nastat s rozvojem technologií pro tvorbu a provoz webových aplikací. Na místo statických stránek, které poskytovaly pouze informace, se začali objevovat služby umožňující daleko větší interakci s uživatelem. Začaly vznikat složité aplikace běžící na webu. Stránky přestaly být oddělenými zdroji statických informací, ale začaly mezi sebou komunikovat, vyměňovat si navzájem svá data a funkční část, kterou disponují jednotlivé služby. Rozšíření internetu mezi uživatele a Web 2.0 se projevil hlavně tím, že začaly vznikat tzv. „sociální sítě“. Ty umožňují uživatelům spolu komunikovat a sdílet informace. To také vedlo k nutnosti vytvoření příjemnějšího, propracovanějšího a intuitivnějšího uživatelského rozhraní těchto webových aplikací.

Nejvýznačnější charakteristiky, které popisují služby Webu 2.0:

Uživatelsky koncentrovaný design

Design webu, který je tvořen tak, že splňuje různé potřeby uživatele a v určitém rozsahu umožňuje přizpůsobení designu, aby uživatelé co nejvíce vyhovoval. Design je čistší, lze se v něm snadno orientovat a často je založen na AJAXu. Pěkným příkladem uživatelsky koncentrovaného designu je přizpůsobitelná domovská stránka společnosti Google, která se jmenuje iGoogle a umožňuje na stránku umístit různé gadgety nebo data z webových zdrojů.

Web jako platforma

Libovolná metoda přístupu k internetu (různé operační systémy, různé webové prohlížeče) neovlivní aplikace Webu 2.0.

Spolupráce

Spolupráce velkého množství lidí na tvorbě obsahu webové aplikace. Výborným příkladem spolupráce je encyklopedie Wikipedia, na které je vidět obrovský potenciál v rámci spolupráce a tvorby obsahu.

Decentralizace

Dříve byla většina služeb spravována administrátorem a ne automaticky. Ovšem služby Webu 2.0 se snaží vše automatizovat, takže zde není zapotřebí nikoho, kdo by povoloval nebo zamítal žádosti od uživatelů. Příkladem je třeba Google AdSense (platforma pro zobrazování reklam na webu). Uživatelé této služby mají k dispozici automatizovaný systém, který jednoduše umožní uživateli vložit reklamu na jeho webové stránky nebo blog.

Software jako služba - SaaS

S velkým rozvojem architektury cloudu se stále více webových služeb snaží jít cestou SaaS. Takto provozovaný software je dostupný jako webová služba, která není závislá na žádné konkrétní platformě.

RIA (Rich Internet Application)

Rich Internet Application je webová aplikace, která se v mnohém podobá desktopovým aplikacím. Často jsou dostupné přes zásuvné moduly webových prohlížečů. Nejrozšířenější platformy pro tvorbu RIA jsou Flash od firmy Adobe, Silverlight od firmy Microsoft a Java od firmy Oracle. [2]

2.3 Mashup

Mashup, ve významu vývoje webu, je webová stránka nebo aplikace, která kombinuje funkcionalitu a datové zdroje ze dvou nebo více zdrojů k vytvoření nové služby. Mashupy pro kombinování dat a funkcionality využívají API (Application Programming Interface), které poskytují velké servery. V tomto případě se jedná o webové API. API je rozhraní, pomocí kterého je možno přistupovat k datům a funkcím jiných webových aplikací. Vývojáři tyto zdroje buď kombinují, nebo k nim přidávají další rozšiřující funkcionalitu.

Využívání API není žádná novinka, ale s příchodem Webu 2.0 umožnila využití služeb poskytovaných webovými aplikacemi, jak to předtím nebylo možné. Mohou vznikat webové aplikace, které by jinak byly velice finančně nákladné nebo jsou příliš specificky zaměřené. Vývojář vytvářející novou službu nemusí znovu implementovat něco, co je již na webu vytvořeno a přístupné přes API. Může proto více času věnovat své aplikaci, což zrychluje a zefektivňuje vývoj a vytvořená služba se může rychleji dostat na trh. Takto může vývojář bez vynaložení většího množství finančních prostředků vytvořit projekt, který by jinak nevznikl. Tyto projekty jsou pro firmy, disponují daty a funkcionalitou potřebnou pro jeho vytvoření, nezajímavé a menší vývojář, který by tento projekt vytvořit mohl, zase nemá prostředky pro jeho realizaci.

Mezi první zdroje pro mashupy patřily API pro mapy (Google Maps, Yahoo, mezi českými zástupci Seznam a Atlas). Velice rychle se také uchytily mashupy, které shromažďují novinky na internetu např. www.delicious.com. Na Delicious uživatelé vkládají odkazy na aktuality a ty jsou zde řazeny na základě hodnocení uživatelů a tagů.

Výhoda poskytování API ke službám je v tom, že je to levná cesta jak poznat, který projekt má potenciál dále se rozvíjet a který ne. Firmy zveřejní rozhraní ke svým službám a nechají menší vývojáře tvořit projekty. Pokud se některý z projektů na trhu uchytí, může ho firma odkoupit a integrovat do svých služeb. Například webová služba Picnic pro online editaci fotografií, která vznikla v roce 2005 a využívala API služby Google Picasa, koupila firma Google v roce 2010 a

integrovala jej do své aplikace Picasa [18]. Výhodou mashupů je, že vývojář který „slepil“ webovou aplikaci za pomoci API se dále nemusí starat o instalaci, aktualizaci a opravování chyb nebo čehokoliv jiného ve využívaných službách [4].

Architektura mashupu se dá rozdělit do tří vrstev [5]:

- Interakce s uživatelem: uživatelské rozhraní mashupu. Technologie: HTML/XHTML, CSS, Javascript, AJAX, Silverlight.
- Služba na webu: zprostředkovává funkcionalitu služby přes API. Technologie: XML-RPC, SOAP, REST.
- Data: manipulace s daty (přijímání, odesílání a ukládání dat). Technologie: XML, JSON, KML.

2.3.1 REST

REST (Representational State Transfer) je architektura rozhraní, kterou v roce 2000 vytvořil Roy Fielding, což je jeden z autorů specifikace protokolu HTTP [6].

Princip této architektury spočívá v tom, že webová služba definuje procedury, které na ní běží a protokol pro volání těchto procedur. Každá procedura má definovanou vlastní URL přes níž se k ní přistupuje [7].

Každá služba poskytující API pomocí architektury REST zpřístupní tyto procedury přes jednotné URL např. `http://api.flickr.com/services/rest/`. Do URL adresy se přidá parametr obsahující jméno procedury, od níž budeme požadovat odpověď a další parametry, které daná procedura požaduje. Jako případ požití uvedu proceduru, kterou poskytuje služba Flickr pro otestování funkčnosti komunikace přes API. Jako volaná procedura bude uvedena `flickr.test.echo`, která při správném fungování vrátí hodnotu uvedenou v parametru `value`. Celá URL dotazu pak bude vypadat takto:

```
http://api.flickr.com/services/rest/?method=flickr.test.echo&name=x
```

Jako odpověď bude poslán jednoduchý XML soubor, v němž je uvedeno, že vše proběhlo v pořádku nebo že došlo k chybě. Pokud došlo k chybě je zde číslo chyby a chybová zpráva, s níž můžeme v dokumentaci k API vyhledat, co chybu způsobilo a jak ji odstranit. XML soubor s odpovědí obsahuje element `rsp`, který má atribut `stat`. Ten nabývá hodnot `ok` nebo `fail` právě podle úspěšnosti dotazu [8]. V tomto projektu je tato architektura použita pro komunikaci se službou Flickr.

3 Použité technologie

Tato část bude obsahovat popis použitých technologií pro vytvoření programové části této práce. Pro uživatelskou část aplikace jsem se rozmýšlel mezi technologií Flash od firmy Adobe a Silverlight od firmy Microsoft, což jsou v podstatě konkurenční technologie. Nakonec byla po konzultaci zvolena technologie Silverlight. Efekty, které se budou moci aplikovat na obrázky použité v komiksu, budou zpracovány pomocí Pixel Shaderu dostupného přes Direct3D API. Pro přístup k datům na službu Picasa je potřeba technologie na serveru, která tuto komunikaci zajistí. Kvůli hostingu, kde aplikace běží, byla zvolena technologie PHP.

3.1 Silverlight

Většina webových stránek byla dříve založena na bezstavovém protokolu HTML. Při tvorbě interaktivních stránek bylo neustále zapotřebí posílat a aktualizovat stavové údaje mezi serverem a klientem a opačně. Toto však aplikaci do značné míry omezuje a zpomaluje ji.

Možností jak tento problém vyřešit je několik. Asi jedno z nejjednodušších řešení je založeno na tom, že se internetový prohlížeč rozšíří o interaktivní prezentační vrstvu (Rich Presentation Tier). Tato nová vrstva zapouzdřuje komunikaci jednotlivých prvků pomocí asynchronních zpráv, které zaručí, že při provádění nějaké akce nezablokuje pracující vlákno a uživatel může nadále pracovat s aplikací. Výsledek je zobrazen až po jeho úplném provedení na serveru a vrácení odpovědi.

Technologie Silverlight značně rozšiřuje interaktivní prezentační vrstvu o možnosti využívání multimédií a vektorové grafiky při tvorbě interaktivních aplikací. Silverlight je nástroj pro tvorbu interaktivních webových aplikací. Ke svému běhu potřebuje zásuvný modul (plug-in). Svou funkcionalitou a zaměřením se hodně podobá technologii Adobe Flash, která je jeho hlavním konkurentem na poli interaktivních webových aplikací.

Běhové prostředí Silverlightu je v podobě zásuvného modulu dostupné pro všechny hlavní webové prohlížeče na operačním systému Windows a Mac OS X. Existuje i varianta pro operační systém Linux. Jedná se o zásuvný modul Moonlight od společnosti Novell vyvíjený ve spolupráci se společností Microsoft. Je zde ovšem menší problém, protože je aktuální stabilní čtvrtá verze Silverlightu a v aktuálním Moonlightu je naimplementovaná podpora Silverlightu ve verzi 3, takže může nastat problém se zobrazením některých nových komponent, které byly přidány až od verze 4 [11]. Microsoft zvolil technologii Silverlight i jako hlavní vývojový prostředek pro jejich novou mobilní platformu Windows Phone 7.

3.1.1 Popis technologie Silverlight

Silverlight je podmnožinou technologie WPF (Windows Presentation Foundation). Hlavní rozdíl mezi Silverlightem a WPF je v tom, že Silverlight ke svému běhu potřebuje pouze zásuvný modul do internetového prohlížeče na rozdíl od WPF. WPF pro svůj běh potřebuje .NET Framework, který je mnohonásobně větší než zásuvný modul. Dříve se také odlišovali tím, že Silverlight byl určen pouze pro běh na webu a WPF byl určen pro běh na lokálním počítači. Silverlight je od verze 3.0 schopen fungovat i v režimu „out of browser“. To znamená, že je možné aplikaci vytvořenou v Silverlightu nainstalovat a provozovat také na lokálním počítači.

Odděluje od sebe grafickou prezentaci uživatelského rozhraní a aplikační logiku aplikace. Uživatelské rozhraní aplikace je umístěno v XAML (Extensible Application Markup Language) souboru. Je to v podstatě XML soubor. Obsahuje stromovou strukturu jednotlivých prvků, které jsou v uživatelském rozhraní obsaženy, definici jejich vlastností a také informace o tom, jak jsou v rámci stránky rozmístěny.

Kód běžící na pozadí Silverlight aplikace, starající se o její chod, je psán v některém z .NET jazyků (C#, Visual Basic). Oddělení grafického rozhraní od aplikační logiky usnadňuje vytváření aplikací. Grafik může vytvořit uživatelské rozhraní a nemusí se zabývat funkcionalitou. Poté programátor k tomuto grafickému návrhu naimplementuje kód na pozadí, který aplikaci dá potřebnou funkcionalitu. Uživatelské rozhraní se dá vytvořit i v aplikačním kódu. To však do značné míry omezí výhody odděleného kódu [11].

Po zkompilování projektu vznikne soubor s příponou .xap v němž jsou data potřebná pro běh aplikace. Tento soubor se spolu se základní webovou stránkou, která automaticky vytvořena, uloží na web, kde k němu mohou přistupovat uživatelé. Při přístupu k aplikaci se nejprve stáhne XAP soubor a zásuvný modul Silverlightu aplikaci z tohoto souboru spustí. Pokud potřebuje aplikace v Silverlightu přistupovat k datům na jiné doméně, než na které je tato aplikace provozována, musí se řídit bezpečnostními pravidly, které Silverlight používá kvůli zabezpečení komunikace mezi různými doménami. Více je zabezpečení komunikace mezi doménami rozebráno zde [10].

3.1.2 Vývojové nástroje

Microsoft pro vývoj aplikací na platformě Silverlight má dva základní nástroje. Jsou to návrhové prostředí Microsoft Expression Blend a vývojové prostředí Visual Studio. Důvodem, proč Microsoft vydal tyto dva nástroje, které se do značné míry překrývají co se týče funkcionality, je ten, že Blend slouží především pro tvorbu grafického uživatelského rozhraní a velice usnadňuje tvorbu animací, efektů a práci s grafikou. Takže designér navrhne uživatelské rozhraní a design stránky a ten je uložen do XAML souboru. Po dokončení návrhu jej předá vývojáři, který používá Visual Studio. Ten do kódu na pozadí, který vykonává aplikační logiku aplikace, dopíše potřebnou funkcionalitu, ale nijak

nezasahuje do již vytvořeného grafického návrhu. Visual Studio poskytuje mnoho funkcí pro zjednodušení psaní kódu a hlavně vynikající ladící nástroje pro odhalování chyb.

3.1.3 Historie technologie Silverlight

První verze Silverlightu byla vydána v roce 2007, takže se jedná o poměrně mladou technologii. Byla vyvíjena pod kódovým označením Windows Presentation Foundation/Everywhere (WPF/E). Obsahovala základní prezentační jádro, které umožňovalo zobrazení uživatelského rozhraní, interaktivity a uživatelský vstup. První verze používala jako programovací jazyk Javascript. Ovládání jednotlivých prvků obstarával skriptovací jazyk, který k těmto prvkům přistupoval pomocí DOM (Document Object Model).

Zároveň s verzí 1.0 vyšla i alfa verze 1.1, jež byla později přejmenována na verzi 2.0. Verze 2.0 již ovšem jako programovací jazyk nepoužívala Javascript, ale bylo možné používat v plné míře .NET jazyky. Silverlight 2.0 také přinesl možnost využívat více než 30 hotových komponent uživatelského rozhraní.

V roce 2009 vyšla verze 3.0, která nově přinášela hardwarově akcelerované video s kodekem H.264 a 3D transformace 2D objektů. Tyto transformace jsou rovněž hardwarově akcelerované, pokud je to možné. Přibyla také možnost vytváření vlastní efektů pomocí využívání pixel shaderů, tyto efekty je možno psát v jazyce HLSL. Také se zde poprvé objevila možnost navázání určité hodnoty nějakého vizuálního prvku na jiný vizuální prvek. Přibyla možnost načítání souboru z disku počítače.

Silverlight ve verzi 4.0 vyšel v polovině roku 2010. Zde přibyla podpora pro internetový prohlížeč Google Chrome, API zpřístupňující webkameru a mikrofon počítače, pokročilé animační efekty, přístup ke schránce operačního systému a mnoho dalšího.

Na podzim roku 2011 by měla vyjít verze 5.0, která je v současné době (duben 2011) dostupná v beta verzi pro vývojáře.

3.2 Pixel shader – HLSL

Efekty použité v tomto projektu jsou psány pomocí jazyka HLSL (High Level Shading Language), který Microsoft uvolnil pro používání pixel shaderu za pomoci Direct3D API. Jazyk HLSL se velmi podobá jazyku C.

Shader je speciální procesor pro zpracování informací přímo grafickou kartou. Pixel shader vezme obrázek na vstupu a provede danou transformaci pro každý pixel obrázku. Tato transformace je naprogramována v jazyce HLSL a uložena do souboru s příponou .fx. Soubor je přeložen překladačem do instrukcí assembleru, aby mohl být zpracován přímo grafickou kartou.

Pro překlad programů je potřeba DirectX SDK, jež obsahuje Direct3D Shader Compiler (program fxc.exe). Operace s jednotlivými pixely jsou velice náročné na hardware, proto

jsou velice dobře optimalizovány. V tomto projektu používám verzi pixel shaderu 2.0, ta je dostupná od DirectX 9.0 [13].

3.3 PHP

PHP je skriptovací jazyk určený pro tvorbu dynamických webových stránek. PHP skripty jsou uloženy na serveru. Při požadavku na server je skript, který je volán, zpracován a vrácen je pouze výsledek operace. PHP bylo vytvořeno v roce 1994, základy má v programovacích jazycích C, Perl, Pascal a Java. PHP je volně šiřitelný software [14].

V tomto projektu je PHP použito, protože Silverlight má bezpečnostní omezení, která mu nedovolí přímo přistupovat na servery Google přes API Picasa. Takže Silverlight aplikace zašle dotaz php skriptu běžícím na webovém serveru, který má nastavenou politiku přístupů mezi doménami tak, že k nim je možno z aplikace Silverlight přistoupit. PHP skripty zprostředkují odpověď od služby Picasa.

4 Návrh aplikace

V této kapitole popisují podrobněji návrh aplikace a uživatelského rozhraní.

4.1 Komiks a jeho ovládání

Hlavním účelem této aplikace je umožnit vytvořit uživateli komiks. Komiks se skládá z několika panelů, ve kterých jsou umístěny obrázky tvořící dohromady určitý příběh.

Při tvorbě byl kladen důraz na to, aby uživatel mohl tento komiks co nejjednodušeji. Po spuštění aplikace je zobrazen panel pro vytvoření nového komiksu. Tlačítko pro vytvoření nového komiksu je umístěno v levém horním rohu aplikace, aby při tvorbě komiksu nerušilo a přitom bylo neustále dostupné.

Při vytvoření schématu komiksu je zapotřebí zadat výšku a šířku výsledného obrázku, který bude obsahovat hotový komiks a počet sloupců a počet řádků komiksu. Výška a šířka schématu nesmí být menší než 300 pixelů. Pokud je zvolena menší velikost, komiks může být špatně čitelný. Schéma vzorového komiksu je zobrazeno na obrázku 4.1



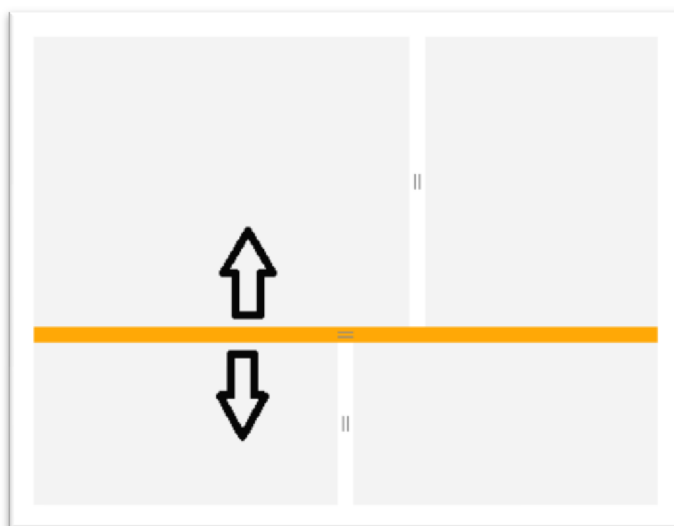
Obrázek 4.1 - schéma vzorového komiksu

Při prvním spuštění je přes schéma zobrazen text s nápovědou, aby bylo uživateli jasné, že pro práci s komiksem je potřeba kliknout na jeden z panelů. Při testování aplikace jsem zjistil, že uživatelé mají problém se zjištěním, jak pokračovat v práci, proto jsem tam umístil nápovědu. Po kliknutí na panel v komiksu se zobrazí náhled obrázku ze zvoleného panelu a menu pro vložení obrázku a jeho editaci. Text nápovědy zobrazený přes schéma komiksu se po prvním výběru panelu skryje a již se nezobrazí. Pouze pokud by uživatel vytvořil nové schéma komiksu.

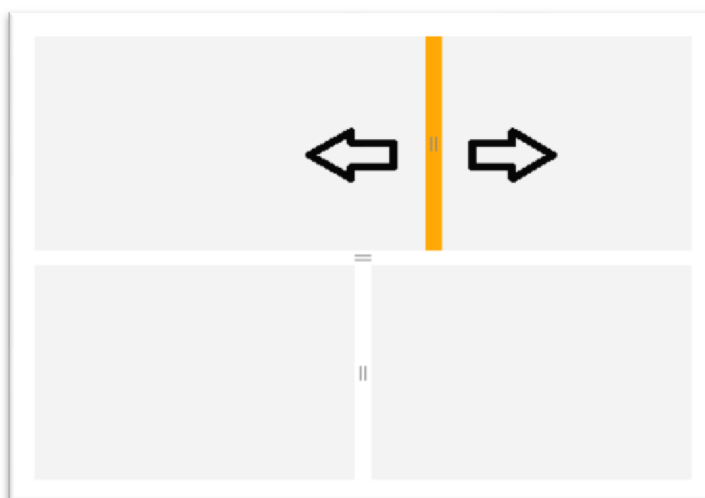
Po dokončení práce s komiksem a dokončení celého příběhu si může uživatel výsledek uložit jako obrázek ve formátu PNG. Uložení probíhá přes ikonu „Uložit komix“ na hlavní straně. Toto tlačítko vyvolá dialogové okno pro uložení komiksu na disk počítače.

4.1.1 Posuvníky

Bílé pruhy mezi jednotlivými panely se dají posunovat a zmenšovat nebo zvětšovat tak prostory pro jednotlivé panely. Horizontální posuvníky posunují všechny panely, které jsou v řádku nad posuvníkem a pod posuvníkem (obr. 4.2). Vertikální posuvník ovlivňuje pouze panely ležící ve stejném řádku, kde se posuvník nachází. Neposunuje se tedy celý blok panelů před celé schéma komiksu jako je tomu v případě horizontálního posuvníku (obr. 4.3).



Obrázek 4.2 - horizontální posuvník



Obrázek 4.3 - vertikální posuvník

4.1.2 Pozice obrázku v panelu

I přes možnost úpravy panelů popsané výše (viz. 4.1.1) nemusí být obsah panelu podle představ uživatele. Při umístění kurzoru myši na konkrétní panel ve schématu komiksu se zobrazí nad panelem komponenta s možností přepnutí pozice zobrazeného obrázku.

Obsahuje tři možnosti nastavení pozice. První z možností je „Vypnit“, při tomto nastavení se obrázek rozprostře po celé ploše panelu, v němž je zobrazen. Tato možnost nezachovává poměr stran obrázku, což může způsobit deformaci obrázku. Druhou možností je „Roztáhnout“. Tato možnost zachová poměr stran obrázku, ale obrázek se zvětší tolik, aby kolem obrázku nezůstalo žádné volné místo. Může to však způsobit, že část obrázku bude mimo panel, takže tato část nebude vidět. Poslední možností je „Přizpůsobit“. Zachová poměr stran obrázku a změní jeho velikost, aby se celý vešel do panelu schématu. Kolem obrázku se může zobrazit prázdné místo, které vyplní zbytek panelu. Použití těchto možností je ukázáno na obrázku 4.4.



Obrázek 4.4 - ukázka možností pozice obrázku ve schématu komiksu

4.2 Výběr obrázku a jeho editace

Po výběru konkrétního panelu v komiksu je umožněno vybrat obrázek do tohoto panelu nebo případně ho změnit, pokud již nějaký obrázek obsahuje. Obrázky lze vybírat z několika zdrojů, které jsou popsány níže. Na obrázky se mohou aplikovat efekty. Každý panel může obsahovat libovolné množství bublin s textem, který vypráví příběh komiksu.

Panel pro výběr a editaci obrázku bude obsahovat prostor s náhledem obsahu aktuálně vybraného panelu (obrázek, kresba, bubliny), menu pro výběr obrázku z konkrétního zdroje, nabídku pro editaci konkrétního obrázku, nabídku pro kreslení do obrázku a vložení a úpravu bubliny.

4.2.1 Zdroje obrázků

V názvu práce je přímo uvedeno slovo „Mashup“. Jak jsem uvedl dříve (viz. 2.3), jedná se využívání zdrojů dat jiných služeb přes jejich API. V tomto případě se jedná o služby, které poskytují své zdroje obrázků. Zvolené služby jsou webová alba Picasa od společnosti Google a Flickr od společnosti Yahoo!. Využitím těchto služeb se uživatel dostane k velkému množství obrázků a fotek jiných uživatelů těchto služeb, kteří je veřejně zpřístupní nebo svých vlastních fotek nahraných na tyto služby. Dále je možné obrázky nahrávat do aplikace přímo z disku počítače, na němž aplikace běží nebo z webové kamery, pokud je jí počítač vybaven.

4.2.1.1 Nahrání obrázku z lokálního disku počítače

Po výběru nahrání obrázku z pevného disku počítače se zobrazí dialog, který dovolí uživateli vybrat si obrázek. Formát obrázku vybraného do komiksu je omezen na formáty JPEG a PNG. V případě pokusu nahrát jiný soubor, který není obrázek, by došlo k chybě při zobrazení. Proto je zde toto opatření.

4.2.1.2 Picasa

Picasa je služba za níž stojí firma Google. Jedná se webové úložiště, kam uživatelé mohou nahrávat své fotografie. Je to jednouchý systém pro publikování fotografií veřejnosti nebo svým známým. Funguje jako úložiště, kam si můžete fotografie zálohovat. Google poskytuje každému uživateli 1GB úložného prostoru zdarma, v případě potřeby lze tento prostor navýšit po zaplacení ročního poplatku až na 16TB. Také je k dispozici program Picasa, který umožňuje jednoduchou editaci fotografií, jejich nahrání na servery Google, vytvoření videa z fotografií nebo třeba odeslání obrázku emailem.

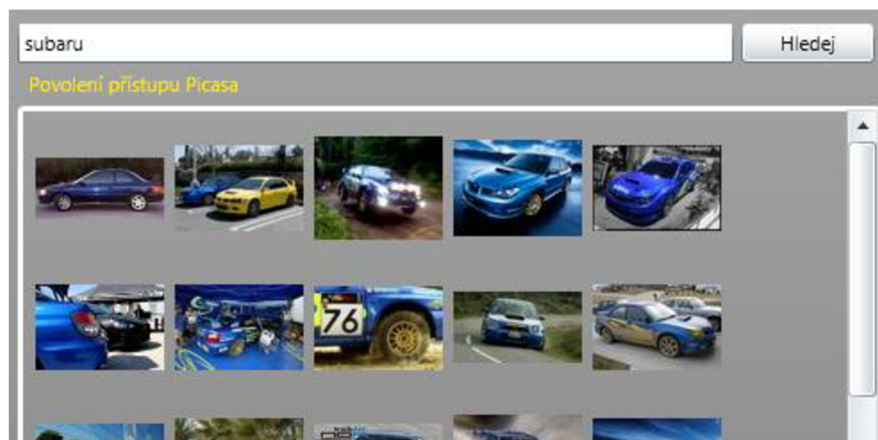
Webová alba fotografií Picasa jsou rozděleny do třech možností přístupu k těmto albům. První možností je „Veřejně dostupné na webu“. Takto označená alba si může kdokoliv prohlížet a fotografie z těchto alb se zobrazují i při vyhledávání přes webový vyhledávač obrázků např. Google Images, Bing Images, ...

Druhou možností je „Všichni uživatelé s odkazem“. Toto nastavení omezí přístup pouze uživatelům, kteří mají odkaz na danou fotografii nebo album. Odkaz na takto zabezpečenou fotografii obsahuje autorizační klíč [9], jenž obsahuje kombinaci čísel a znaků, takže je těžké jej uhodnout v případě, že by se k fotografii chtěl dostat uživatel bez odpovídajícího odkazu.

Třetí a poslední možností je „Soukromé“. To je nejvyšší úroveň zabezpečení. Uživatel vlastníci toto album přesně definuje, kteří uživatelé budou mít k tomuto albu přístup a tito uživatelé mají udělen přístup pouze, pokud jsou přihlášení přes účet Googlu pro potvrzení své identity.

Při výběru obrázku z webových alb Picasa bude mít uživatel možnost vyhledávat ve veřejných albech, do kterých mají přístup všichni uživatelé. Vyhledávání je založeno na vyhledání fotografií podle konkrétních jmen fotografií. Zobrazí se prvních 100 fotografií, které na dotaz přes

API vrátí služba Picasa (obr. 4.5). Po kliknutí na konkrétní obrázek načtený v boxu pod vyhledávacím polem se obrázek zobrazí v poli pro náhled, kde se na něj aplikují efekty, kresba a vkládají se bubliny.

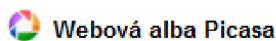


Obrázek 4.5 - ukázka vyhledání obrázku na Picasa podle klíčového slova "subaru"

Kromě vyhledávání ve veřejných galeriích je zde i možnost, že povolí aplikaci přístup k jeho webovým albům a to jak k veřejným, tak i k soukromým. Na obrázku 4.5 je pod polem, kam se zapisuje vyhledávaný pojem, vidět odkaz „Povolení přístupu Picasa“. Po kliknutí na tento odkaz je otevřena nová karta webového prohlížeče. V této kartě je zobrazena stránka, kde se uživatel přihlásí na svůj Google účet, pokud už není přihlášený (když uživatel nemá u Googlu účet, může si ho zde vytvořit). Poté se zobrazí stránka, na níž je potřeba povolit aplikaci přístup k potřebným datům přihlášeného uživatele. Jedná se pouze o povolení přístupu k datům webových alb Picasa. Uživatel se nemusí obávat, že by aplikace mohla přistupovat k jiným službám, které Google poskytuje. Povolení se týká pouze jedné konkrétní služby, jak je vidět na obrázku 4.6. Aplikace nebude mít přístup k osobním údajům uživatele (kromě jména a příjmení) a ani k jeho heslu.



Web picasa.area59.eu žádá o přístup k vašemu účtu Účet Google pro následující produkty.

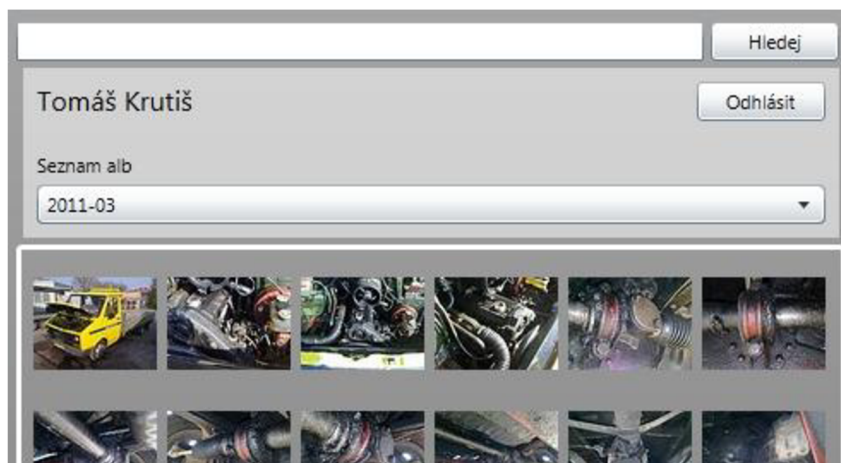


Obrázek 4.6 - příklad povolení přístupu ke službě Picasa

Po povolení přístupu k webovým albům se stránka, kde se přístup přiděloval, přesměruje na stránku aplikace, na níž se zpracují potřebná data pro komunikaci s Picasa a uloží se pro pozdější potřebu. Tuto stránku je potřeba zavřít (stránka obsahuje odkaz na zavření stránky) a pokračovat na stránce aplikace, kde byl stisknut odkaz „Povolit přístup Picasa“.

Po zavření stránky povolení přístupu se na stránce aplikace zobrazí místo odkazu na povolení přístupu tlačítko „Přihlásit Picasa Google“. Zobrazí se panel se jménem uživatele, tlačítko pro odhlášení přihlášeného uživatele, seznam alb uživatele se vypíše do rozbalovacího seznamu a

načte se obsah prvního alba, který je první v seznamu alb (obr. 4.7). Pokud nemá uživatel ve webových albech uloženy žádné fotografie, tak se přihlásí, ovšem nenačtou se žádné fotografie.



Obrázek 4.7 - ukázka přihlášení uživatele do webových alb Picasa

Po rozbalení seznamu alb si uživatel může vybrat libovolné svoje album a kliknutím na toto album zobrazí jeho obsah v seznamu obrázků pod seznamem alb. Kliknutím na konkrétní obrázek v albu se tento obrázek zobrazí v panelu s náhledem obrázku. Potom se obrázek může editovat, kreslit do něj nebo vkládat bubliny.

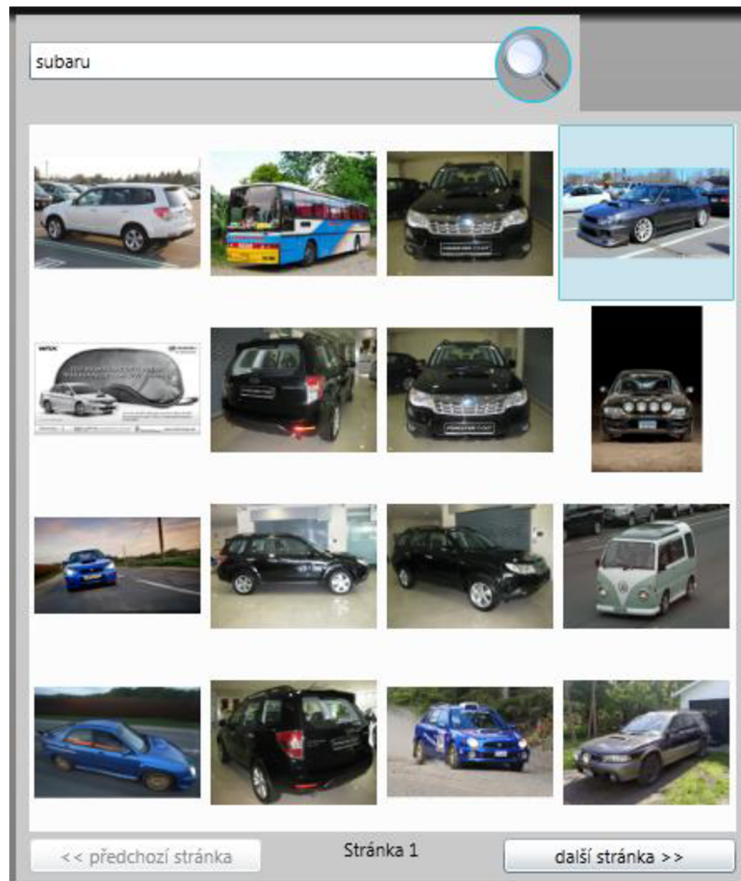
Po přihlášení může uživatel stále vyhledávat ve fotografiích, ale stále vyhledává ve veřejných galeriích všech uživatelů a ne pouze ve svých albech.

4.2.1.3 Flickr

Flickr je služba velmi podobná Picase ovšem je provozována firmou Yahoo!. Uživatelé sem také mohou nahrávat své fotografie. V aplikaci „Generátor komixů - Mashup“ je možno na Flickru vyhledávat obrázky podle zadaných slov. Vyhledávání řadí na přední pozice nejnovější fotografie uživatelů, takže výsledky se neustále mění podle aktuálně nahraného obsahu. Opět se vyhledává pouze ve veřejných galeriích.

Fotografie mohou mít několik možností nastavení zabezpečení. Mohou být privátní, takže k nim může přistupovat pouze vlastní této fotografie. Nebo mohou být zařazeny do skupiny „Your Friends“ a „Your Family“, to je zpřístupní uživatelům, které má vlastník v těchto skupinách zařazeny. Jako poslední možnost nastavení zabezpečení je „Anyone - public“, to zpřístupní fotografii všem a je možno ji najít i ve vyhledávací obrázků.

Při vyhledání v mojí aplikaci se zobrazí prvních 16 získaných obrázků z Flickru. Pod polem s náhledy jsou k dispozici tlačítka pro přechod na další 16 obrázků. V podstatě je vrácený seznam obrázků rozdělen na stránky, v tomto případě je na jedné stránce 16 obrázků. Je potřeba zavést stránkování obrázků, protože Flickr by bez omezení vrátil data pro 4000 vyhledaných obrázků. Pro přehledné zobrazení a pro rychlejší načítání je potřeba obrázky rozdělit na tyto stránky (obr. 4.8).



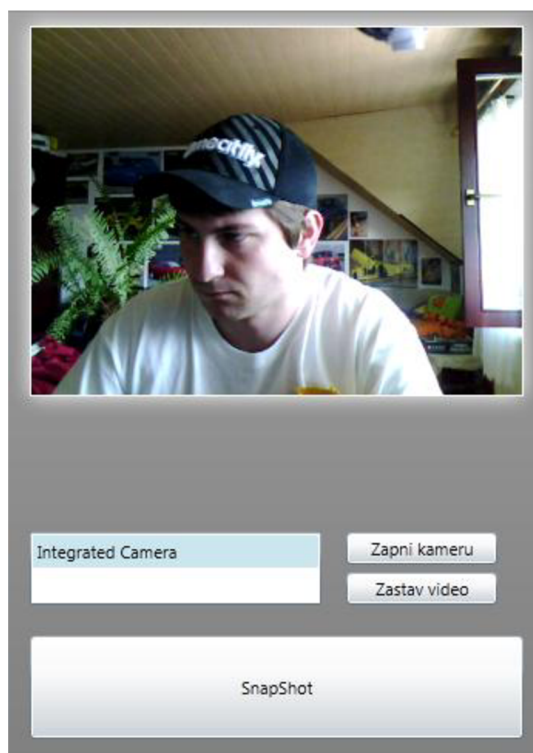
Obrázek 4.8 - vyhledání obrázků na službě Flickr a stránkování obrázků

4.2.1.4 Webkamera

V dnešní době je naprostá většina notebooků vybavena webovou kamerou, takže mi přišlo jako dobrý nápad použít ji při tvorbě komiksu. Při výběru webové kamery jako zdroje obrázku je uživateli zobrazeno menu, jež obsahuje seznam webových kamer dostupných na počítači, na kterém je aplikace spuštěna. Pokud je seznam prázdný, tak k počítači není žádná webová kamera připojena nebo není správně nainstalován ovladač této kamery. Jeli v počítači více webových kamer, může si uživatel zvolit libovolnou z nich.

Po výběru zdroje videa se zapne zobrazení videa tlačítkem „Zapni kameru“ a obraz z kamery se objeví v bílém čtverci nad výběrem zdroje (obr. 4.9). Zároveň se zapnutím obrazu se objeví tlačítko „SnapShot“, kterým se udělá snímek z kamery v okamžik stisku tohoto tlačítka a zobrazí se v panelu s náhledem obrázku do komiksu.

Pro vypnutí kamery je zde tlačítko „Zastav video“, to kryje tlačítko pro sejmutí obrázku a zastaví zobrazování dat z webové kamery.



Obrázek 4.9 - zdroj videa z webové kamery a tlačítka pro obsluhu

4.2.2 Bubliny

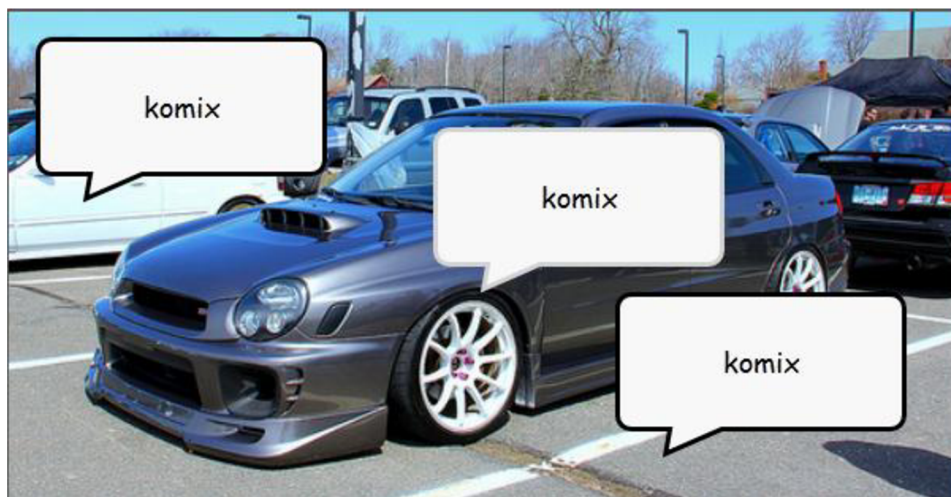
Příběh komiksu kromě samotných obrázků v jednotlivých panelech dokresluje také text, který je v panelech umístěn. Ten je realizován jako bubliny vkládané k obrázkům, aby doplnily příběh komiksu.

Po výběru konkrétního panelu v komiksu pro editaci, je v menu pro výběr a editaci obrázku možnost vložit bublinu do komiksu pomocí tlačítka „Přidat bublinu“. Do každého panelu komiksu je možno vložit libovolné množství bublin. Bublina po vložení obsahuje výchozí text, má výchozí velikost a výchozí velikost písma.

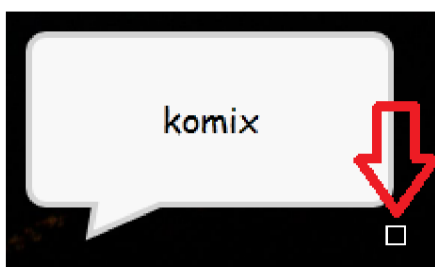
Změna jednotlivých vlastností bubliny se provádí po výběru konkrétní bubliny. Okraj této bubliny změní barvu, aby bylo jasné, která bublina byla vybrána k editaci a zobrazí se menu umožňující změnu vlastností (obr. 4.10).

V menu editace vybrané bubliny se dá změnit text napsaný v bublině, velikost písma textu v bublině, smazat vybranou bublinu a změnit tvar bubliny. Bublina může mít tvar obdélníku, zaobleného obdélníku, oválný tvar a tvar mráčku.

Pro změnu velikosti bubliny slouží malý obdélník v pravém dolním rohu bubliny, který se zobrazí při najetí kurzorem myši nad bublinu. Posunováním tohoto malého obdélníku se bublina zmenšuje nebo zvětšuje. Bublina se zobrazeným obdélníčkem pro změnu velikosti je na obrázku 4.11.



Obrázek 4.10 - šedě orámovaná bublina je vybrána pro editaci



Obrázek 4.11 - červená šipka označuje bod sloužící ke změně velikosti bubliny

4.2.3 Kresba do obrázku

Do obrázku je možno nakreslit jednoduchou kresbu. Kreslit lze přímo do náhledu obrázku vybraného panelu komiksu. V nastavení kresby lze změnit barvu pera, kterým se kreslí, nebo celou kresbu z obrázku smazat. Kresba je vložena až za bubliny, takže kresba nepřekryje text v bublinách a nedojde ke ztrátě obsahu. Příklad kresby je na obrázku 4.12, na němž je v obrázku nakresleno slovo „TOMÁŠ“. Kresbu lze z obrázku odstranit pouze jako celek, nejde smazat pouze část kresby.



Obrázek 4.12 - příklad kresby v obrázku

4.2.4 Efekty

Efekty umožňují upravit obrázek získaný z nějakého zdroje do podoby, která bude uživateli vyhovovat více než původní obrázek. V tomto projektu je k dispozici pět efektů: rozostření, rozvíření, pixelizace, změna barevného tónu a tzv. speed efekt. Efekty se dají vzájemně kombinovat. Každý efekt lze vzít zpět a vrátit tak obrázek do původního stavu. Efekty se aplikují pouze na obrázek v panelu komiksu, kresba v obrázku a bubliny nejsou efekty nijak modifikovány.

4.2.4.1 Rozostření

Rozostří obrázek vybraný do panelu komiksu. Posuvníkem v menu rozostření lze měnit míru rozostření, pokud je posuvník úplně vlevo je hodnota vlastnosti, která nastavuje rozostření nulová, takže efekt obrázek nijak neovlivňuje. Na obrázku 4.13 je vidět, jak efekt modifikuje originální obrázek.



Obrázek 4.13 - efekt rozostření použitý na obrázku

4.2.4.2 Rozvíření

Tento efekt vytvoří vír na upravovaném obrázku. Opět jako v případě rozostření se efekt aplikuje pomocí posuvníku, když je posuvník úplně vlevo, efekt nemá na obrázek žádný vliv. Změna pozice posuvníku doprava určuje míru zatočení víru v obrázku. Ukázka efektu je na obrázku 4.14.



Obrázek 4.14 - efekt rozvíření

4.2.4.3 Pixelizace

Pixelizace vytvoří hrubší strukturu bodů obrázku. Efekt je potřeba zapnout, protože pouhým nastavením posuvníků není možné efekt nastavit tak, aby nijak nemodifikoval originální obrázek. Po zapnutí efektu se zobrazí tři posuvníky. První dva posuvníky udávají počet bodů efektu na výšku a na šířku. Třetí posuvník určuje posunutí horizontální posunutí sudých a lichých řádků bodů. Vzájemné posunutí může být nejvíce o polovinu bodu. Příklad použití efektu je na obrázku 4.15. Při vypnutí efektu tlačítkem „Vypnout“ se obrázek vrátí do původního stavu a posuvníky nastavující vlastnosti efektu se skryjí.



Obrázek 4.15 - efekt pixelizace se vzájemným posunem lichých a sudých řádků

4.2.4.4 Změna barevného tónu

Originálnímu obrázku změní barevný tón. Efekt je potřeba aktivovat tlačítkem „Zapnout“. Po jeho aktivaci se zobrazí panel s barvami, které se mohou u efektu použít. Výchozí barevný tón po zapnutí je červený. Kliknutím na jinou barvu v panelu se barevný tón změní. Na obrázku 4.16 je ukázka změny barevného tónu na zelenou barvu.



Obrázek 4.16 - změna barevného tónu na zelenou

4.2.4.5 Speed efekt

Efekt provede rozmazání obrázku od jednoho centrálního bodu, který je nastavitelný. Po zapnutí efektu se zobrazí tři posuvníky. První dva posuvníky nastavují střed rozostření. Jeden posuvník určuje souřadnice v ose x a druhý v ose y . Třetí posuvník určuje rozsah rozostření od nastaveného centrálního bodu. Použití speed efektu je zobrazeno na obrázku 4.17.



Obrázek 4.17 - speed efekt s výchozím nastavením

5 Implementace

Tato kapitola popisuje některé základní prvky použité při implementaci projektu a popis jejich funkčnosti. Aplikace je vytvořena technologií Silverlight. Grafické rozhraní je naprogramováno pomocí XAML a funkční kód na pozadí aplikace, který zpracovává a vykonává požadavky uživatele je naprogramován v jazyce C#. Při práci s webovými alby Picasa, je potřeba obejít bezpečnostní opatření technologie Silverlight. To je vyřešeno pomocí php skriptů na doméně, kam má aplikace přístup. Zbytek komunikace s API Picasy zajistí skripty, které vrátí odpověď zpět aplikaci. Při programování jsem používal tyto knihy [11], [12] a [13].

5.1 Třída Komix

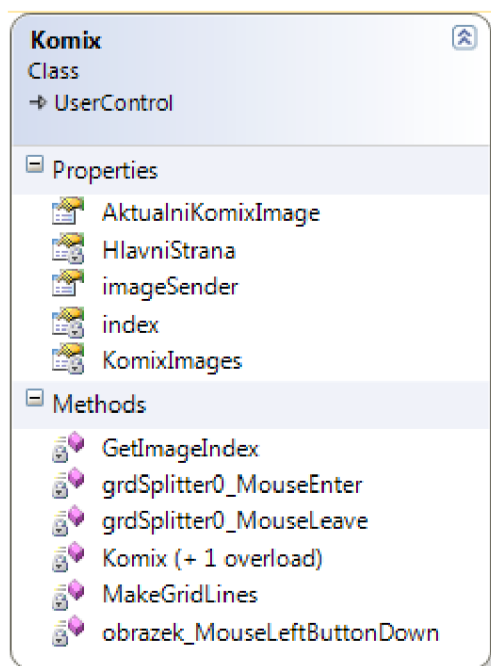
Instance třídy Komix vytvoří nový komiks, který je vytvořen pomocí informací o komiksu zadaných uživatelem při tvorbě nového schématu komiksu. Třída dědí od třídy `UserControl`, takže se jedná o grafickou komponentu, kterou je možné při návrhu vložit přímo do XAML souboru.

Při vytváření objektu této třídy je potřeba vědět, jakou má mít nový komiks šířku a výšku a počet panelů na řádku a ve sloupci. Třída má proto dva konstruktory. První je bezparametrový, inicializuje objekt jako grafickou komponentu a je nastaven jako privátní. Druhý konstruktory je veřejný a jako parametry přijímá požadované parametry pro vytvoření komiksu, další parametr je odkaz na hlavní stranu aplikace, aby objekt komiksu mohl přistupovat ke komponentám na hlavní straně, což je potřeba při kliknutí na konkrétní panel, který chce uživatel změnit. Tvorbu schématu komiksu má na starost funkce `MakeGridLines`. Po kliknutí na panel se zobrazí náhled obrázku z vybraného panelu a menu s výběrem obrázku.

Při vytváření komiksu se nejprve vytvoří jedna komponenta `Grid`, která obsahuje celé schéma komiksu a má nastaveny rozměry, které byly zadány jako parametr šířka a výška komiksu. Hlavní `Grid` se rozdělí rovnoměrně na řádky podle parametru, který určuje počet panelů na výšku komiksu. Mezi řádky je vždy vložen ještě jeden řádek o velikosti 10 pixelů, jenž je zde pro komponentu `GridSplitter`. Tato komponenta při přetažení myši na jiné místo mění poměr okolních řádků s panely (obr. 4.2). Do každého řádku je poté vytvořena komponenta `Grid`, která je rovnoměrně rozdělena podle parametru s počtem panelů v řádku komiksu. Také jako u řádků je mezi jednotlivé sloupce vložen sloupec o velikosti 10 pixelů pro `GridSplitter`, aby bylo možné měnit poměry velikostí sousedních panelů (obr. 4.3).

Po vytvoření schématu komiksu je možné do každého sloupce vnitřních `Gridů` vložit obrázek. Obrázek je reprezentován jako objekt třídy `KomixImage` (viz. 5.2). Pro lepší práci s obrázky

v komiksu je vytvořeno pole objektů s obrázky. Toto pole je indexováno jmény obrázků. Diagram třídy `Komix` je na obrázku 5.1.

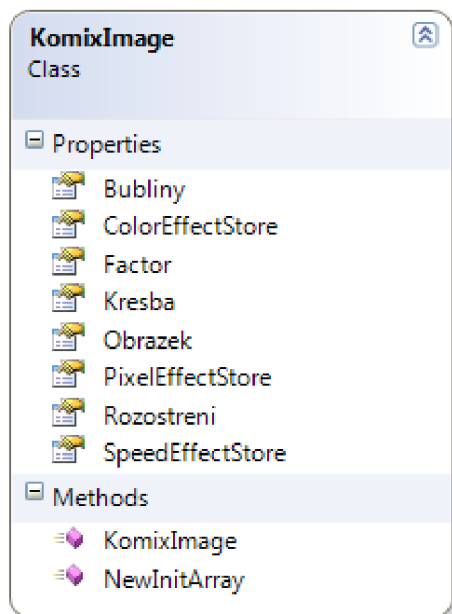


Obrázek 5.1 - diagram třídy `Komix`

5.2 Třída `KomixImage`

Objekt této třídy uchovává informace o jednom konkrétním panelu ve schématu komiksu. Diagram třídy `KomixImage` je na obrázku 5.2. Každý objekt nese informace o všech efektech, bublinách a kresbě v daném panelu. Také je zde uložen obrázek, který byl uživatelem do panelu vybrán nebo je tu obrázek výchozí. Obsahuje také pozici obrázku v panelu. Po vytvoření objektu jsou všechny efekty vypnuty, seznam bublin je prázdný a panel neobsahuje žádnou kresbu.

Třída má statickou metodu pro vytvoření pole o zadané velikosti, které obsahuje objekty této třídy. Metoda se používá při vytváření schématu komiksu, kdy se vytvoří pole pro všechny panely v komiksu.

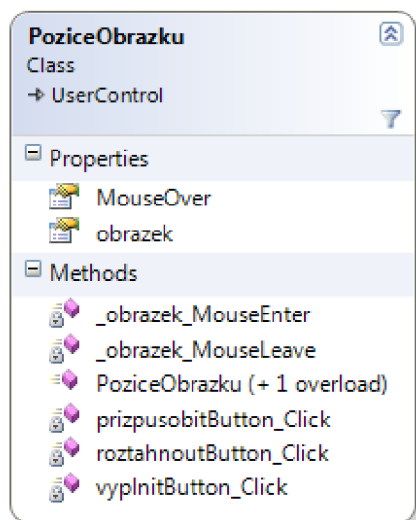


Obrázek 5.2 - diagram třídy KomixImage

5.3 Třída PoziceObrazku

Objekt této třídy se používá na změnu pozice obrázku v panelu komiksu. Diagram třídy je na obrázku 5.3. Třída `PoziceObrazku` dědí od třídy `UserControl`, takže se jedná o grafickou komponentu. Při vytváření schématu komiksu je do každého panelu vložen objekt této třídy, který ovlivňuje obrázek v tomto panelu a podle nastavení uživatele mění jeho pozici. Při vytvoření objektu této třídy je potřeba vložit do vlastnosti `obrazek` odkaz na obrázek v panelu komiksu, který se bude podle výběru pozice měnit (obr. 4.4).

Při přejetí kurzorem myši přes obrázek, který je spojený s objektem této třídy, se vyvolají události definované v této třídě (`_obrazek_MouseEnter` a `_obrazek_MouseLeave`), které komponentu na změnu pozice obrázku v panelu zobrazí nebo skryjí.

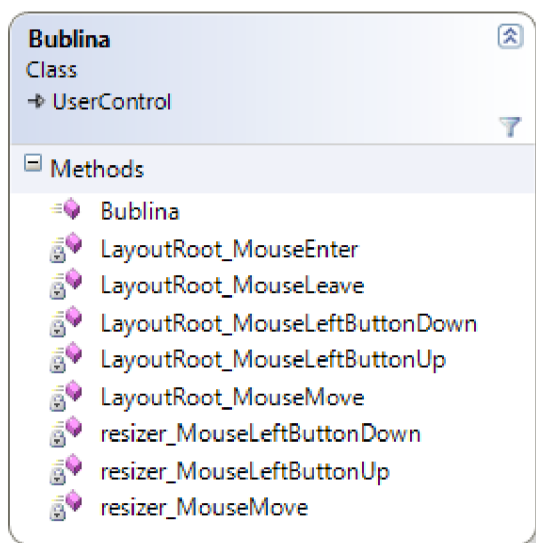


Obrázek 5.3 - diagram třídy PoziceObrazku

5.4 Třída Bublina

Instance této třídy vytvoří komponentu bubliny, která se vkládá do panelu komiksu pro zobrazení textu pro dokreslení příběhu. Třída obsahuje události, které zpracovávají posouvání bubliny po panelu komiksu a zvětšování a zmenšování bubliny pomocí malého čtverečku `resizer`. Změna pozice nebo velikosti bubliny je realizována změnou nastavení odsazení (vlastnost `margin`) každé ze stran bubliny od okrajů panelu.

Objekty bublin jsou vkládány do nejvyšší vrstvy panelu komiksu, takže překryje kresbu v obrázku a všechny aplikované efekty.



Obrázek 5.4 - diagram třídy Bublina

5.5 Třída FlickrPhoto

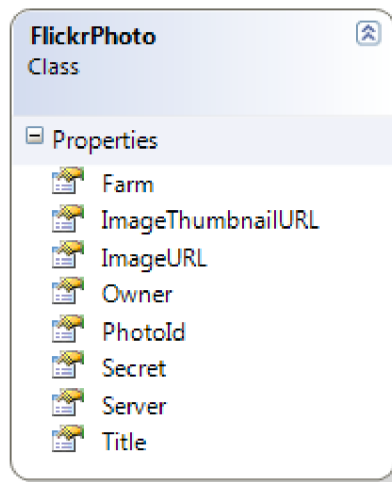
Objekt vytvořený z této třídy uchovává potřebné informace o fotografii z webové galerie Flickr. Na diagramu třídy (obr. 5.5) jsou vidět vlastnosti, kde se tyto informace ukládají. Vlastnost `Title` a `Owner` obsahuje jméno obrázku a jméno uživatele, který fotografii na Flickr uložil. Vlastnosti `Farm`, `Server`, `Secret` a `PhotoId` jsou určeny pro sestavení URL adresy fotografie v plném rozlišení. Tuto URL adresu vrátí zavolání vlastnosti `ImageURL`.

Adresu obrázku lze získat správnou kombinací potřebných vlastností podle následujícího vzoru:

```
http://farm{Farm}.static.flickr.com/{Server}/{PhotoID}_{Secret}.jpg
```

Do složených závorek jsou vloženy uložené vlastnosti.

Při načítání většího množství fotografií je lepší používat náhledy. Stažení náhledů trvá podstatně kratší dobu než stahování fotografií v jejich originálním rozlišení. Vlastnost `ImageThumbnailURL` obsahuje URL adresu náhledu obrázků. Velikost delší strany náhledu je 100 pixelů. Komunikace s API služby Flickr probíhá pomocí architektury REST (kapitola 2.3.1).



Obrázek 5.5 - diagram třídy FlickrPhoto

5.6 Efekty

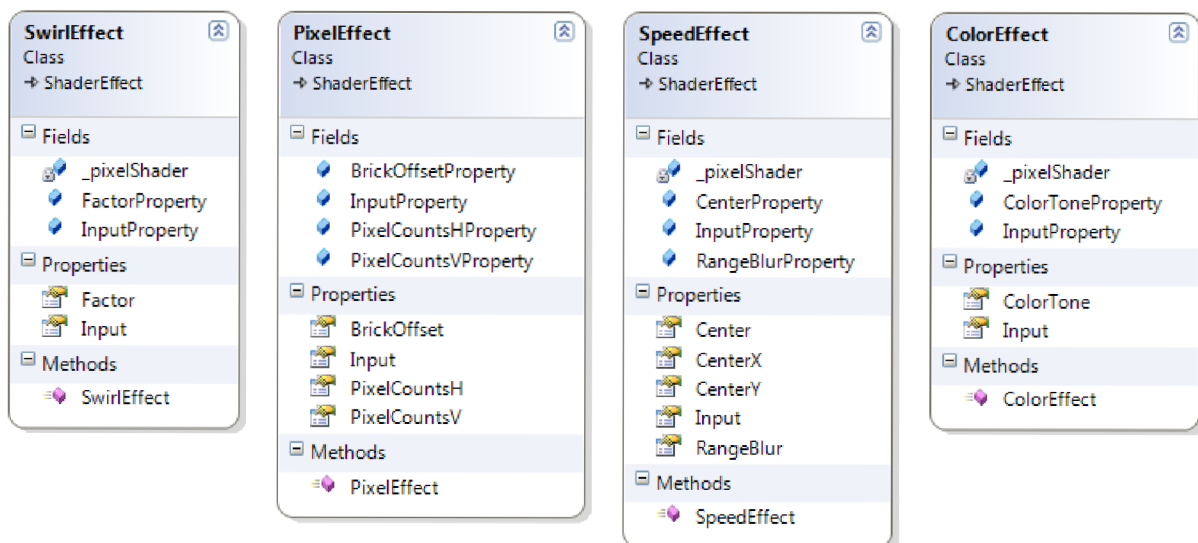
Efekty se používají k modifikaci obrázku v panelu komiksu. Ovlivňují pouze obrázek. Kresba a bubliny v panelu po aplikaci efektu zůstanou nezměněny. Třídy efektů dědí od třídy `ShaderEffect`. To umožňuje tvorbu vlastních efektů s použitím Pixel Shaderu [15][16].

Efekt rozostření použitý v projektu je dostupný knihovně dodávané Microsoftem. Ostatní efekty jsou vytvořeny pomocí Pixel Shaderu. Každý vytvořený efekt má vlastní třídu. Diagramy tříd efektů jsou na obrázku 5.6.

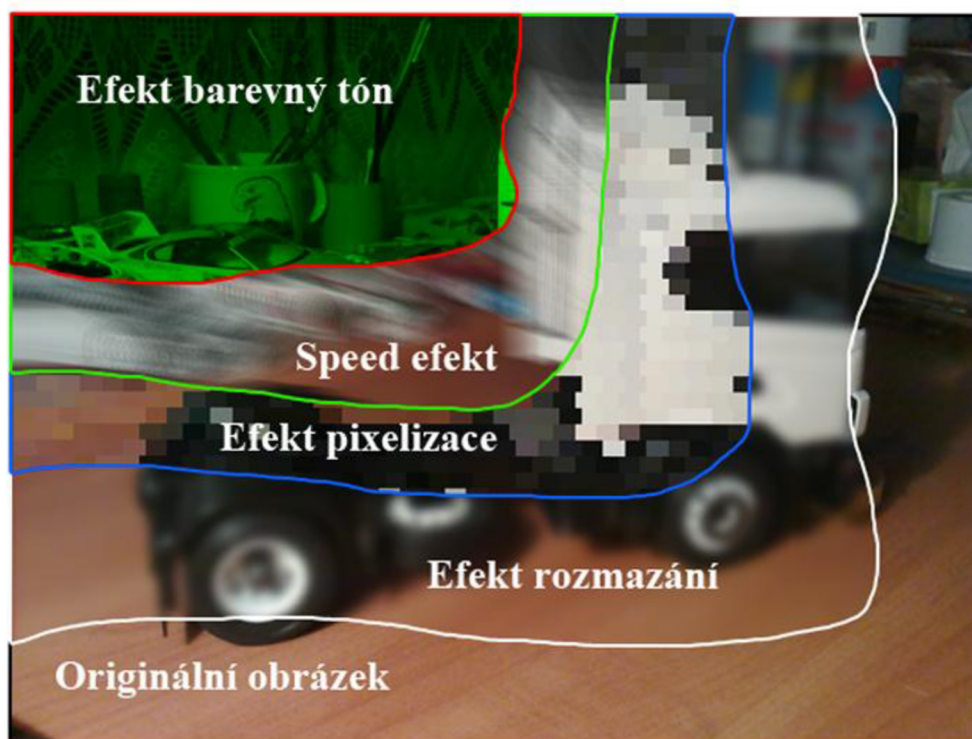
Každý efekt má vlastnost `Input`. Ta předává grafický vstup (`Brush`) na zpracování. Efekt postupně projde všechny pixely a provede s nimi požadovanou transformaci. Každý efekt má ještě další vlastnosti, kterými se nastavují transformace efektů.

Při vzniku nového objektu efektu je vytvořen objekt `PixelShader` odkazující na vytvořený soubor s příponou `.ps`, který obsahuje instrukce pro zpracování efektu na grafické kartě. Tento soubor byl získán přeložením zdrojového kódu efektu napsaného v jazyce HLSL. Jednotlivé vlastnosti pro změnu nastavení efektu jsou spojeny s registry Pixel Shaderu. Spojení vlastnosti s registrem se provádí pomocí metody `DependencyProperty.Register`. Vstup, který má být efektem změněn, je u každého efektu spojen s registrem `S0` a k tomuto registru se přiřadí metodou ze třídy `ShaderEffect`. Metoda se jmenuje `RegisterPixelShaderSamplerProperty`.

Efekty nejdu aplikovat na komponentu `Image` všechny najednou. Aby šli efekty kombinovat, musí se pro každý vytvořit nový rámeček obrázku, na nějž se pak efekt použije. Průřez jednotlivými vrstvami s efekty je na obrázku 5.7. To umožňuje kombinovat efekty mezi sebou, protože rámeček s efektem mění všechny prvky, které obsahuje, takže i ostatní vnořené rámečky s efekty.



Obrázek 5.6 - diagramy tříd efektů



Obrázek 5.7 - vrstvy s efekty nad originálním obrázkem

5.7 Picasa Wrapper

5.7.1 Přístup k webovým albům Picasa

Silverlight aplikace při přístupu na server s jinou doménou než na níž běží, vyžaduje v kořenovém adresáři serveru, od kterého aplikace požaduje data, soubor obsahující nastavení politiky přístupu z jiných domén.

Silverlight akceptuje dva typy souborů s nastavením politiky:

- Soubor určený pro Silverlight aplikace (clientaccesspolicy.xml)
- Soubor určený pro Adobe Flash aplikace (crossdomain.xml)

Pokud kořenový adresář serveru neobsahuje některý s těchto souborů, aplikace se nemůže k této službě připojit a požadovat od ní data. Také pokud v tomto souboru není povolen přístup z domény, ze které se aplikace připojuje nebo není povolen přístup z jakékoliv domény, tak není možné zahájit komunikaci [10].

Právě tento problém nastal u přístupu k API služby Picasa. V kořenovém adresáři `picasaweb.google.com` je sice soubor `crossdomain.xml`, ale ten povoluje přístup pouze z domén Googlu. Takže bylo nutné tento problém vyřešit.

Řešení spočívá v tom, že Silverlight aplikace přistupuje k php skriptům na stejné doméně na které běží. Skripty dotaz předají přes API službě Picasa a po obdržení odpovědi ji zašlou zpět Silverlight aplikaci. Skripty mohou k API normálně přistupovat, protože PHP žádné povolení přístupu nepotřebuje.

Pro přístup z jakékoliv domény je potřeba, aby v kořenovém adresáři domény byl soubor `crossdomain.xml` a v něm bude element `allow-access-from`, který to povolí. Příklad je uveden níže:

```
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

Nebo soubor `clientaccesspolicy.xml`. V něm bude element `allow-from`, který povolí přístup z jakékoliv domény a element `grant-to` pro povolení přístupu do určitých adresářů na serveru:

```
<allow-from http-request-headers="SOAPAction">
  <domain uri="*" />
</allow-from>
<grant-to>
  <resource path="/" include-subpaths="true" />
</grant-to>
```

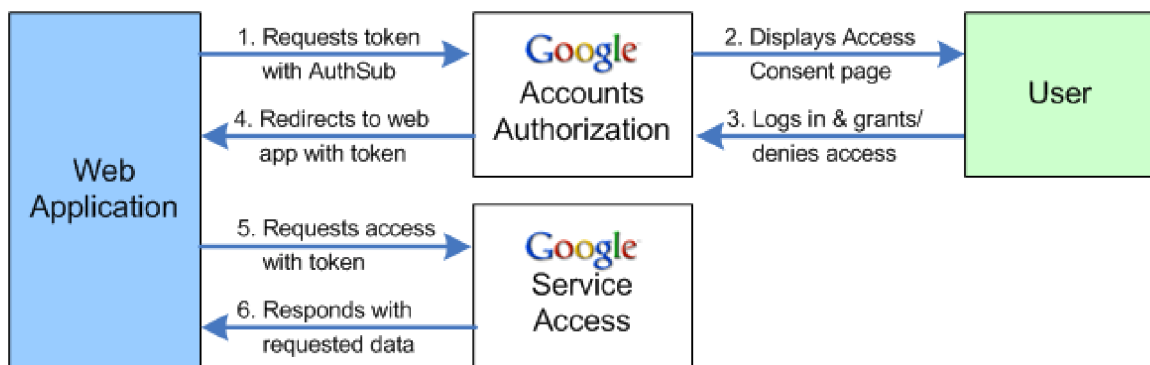
5.7.2 Vyhledávání

Vyhledávání je realizováno pomocí asynchronního zaslání dotazu skriptu `search_picasa.php`. Ten prostřednictvím API požádá službu Picasa o vyhledání obrázků podle zadaného slova nebo slovního spojení. Služba vrátí XML soubor, který obsahuje informace o vyhledaných obrázcích. Tato odpověď je odeslána Silverlight aplikaci.

V aplikaci je XML odpověď rozebrána na informace o jednotlivých obrázcích a je vytvořen seznam objektů třídy `PicasaPhoto` obsahující tyto informace. Objekt `PicasaPhoto` má uložené jméno obrázku, URL adresu obrázku v plném rozlišení a URL adresu náhledu obrázku se šířkou 72 pixelů.

5.7.3 Přihlášení do webových alb Picasa

Přihlášení probíhá v několika krocích. Nejprve je potřeba získat povolení přístupu k datům uživatele ze služby Picasa. Pro webové aplikace, které mohou mít v jednom okamžiku více uživatelů, je určeno rozhraní pro autorizaci AuthSub. Postup autorizace přes AuthSub je vidět na obrázku 5.8.



Obrázek 5.8 - postup při autorizaci přes rozhraní AuthSub (zdroj: Google Code)

Prvně je nutné získat přístup k účtu uživatele. Je zobrazen nový panel prohlížeče s webovou stránkou, na níž uživatel povolí přístup k jeho uživatelským datům. Jedním z parametrů URL adresy stránky s povolením přístupu je parametr `next`, jenž obsahuje adresu stránky pro zpracování tokenu. Token je uložen pro použití při další komunikaci se službou. Obdržený token je ale platný pouze pro jeden dotaz.

Pro získání tokenu pro více dotazů na službu je potřeba pomocí metody `AuthSubSessionToken` poslat již získaný token určený pro jeden dotaz a v hlavičce odpovědi je vrácen nový token, který už zvládne obsloužit libovolné množství dotazů. Každý další požadavek na API služby Picasa musí obsahovat, v HTTP hlavičce požadavku, řádek se získaným tokenem:

```
Authorization: AuthSub token="token"\r\n
```

5.7.4 Alba přihlášeného uživatele

Po povolení přístupu k účtu uživatele a získání tokenu je možné poslat na API požadavek na získání XML souboru s informacemi o albech uživatele. V HTTP hlavičce dotazu musí být uveden řádek s tokenem pro autorizaci uživatele. Zaslání dotazu má na starost skript `get_albums.php`, jenž vytvoří socket, který pomocí SSL zašle HTTP dotaz na port 443. Obdržený XML soubor s odpovědí pouze vrátí aplikaci, která si jej již zpracuje.

Příklad níže ilustruje hlavičku dotazu pro přístup k seznamu webových alb Picasa:

```
GET /data/feed/api/user/default HTTP/1.1
Host: picasaweb.google.com
Content-type: application/x-www-form-urlencoded
Authorization: AuthSub token="získaný token"
Connection: close
```

Jako identifikační číslo v URL adrese se uvedeno slovo `default`, protože uživatel je identifikován pomocí tokenu uvedeného v hlavičce jako položka `Authorization`.

Po obdržení odpovědi, ji skript zašle zpět aplikaci, která přijatá data ve formátu XML analyzuje a vytvoří seznam objektů třídy `PicasaAlbum`. Objekt ukládá o každém albu jeho jméno, URL adresu s odkazem na toto album, přístupová práva k albu, počet obrázků, které obsahuje a odkaz na náhled jednoho obrázku z alba.

5.7.5 Výběr alba

Po načtení všech alb uživatele a výběru jednoho konkrétního alba se z objektu, patřícímu k tomuto albu, načte odkaz na toto album. Odkaz je předán skriptu `get_album.php`. Tento skript funguje stejně jako skript pro získání alb uživatele, ale ve vráceném XML souboru jsou informace o obrázcích z vybraného alba.

Po předání odpovědi zpět aplikaci, je odpověď analyzována a jsou z ní vybrány potřebné informace o obrázcích. Je vytvořen seznam objektů třídy `PicasaPhoto`. Každý objekt má v sobě uloženy informace o jednom obrázku z alba. Konkrétně ukládá název obrázku, URL adresu náhledu obrázku o šířce 72 pixelů a URL adresu obrázku v originálním rozlišení. Náhledy jsou použity při zobrazování obsahu alba, jelikož načítání náhledů je méně časově náročné než načítání obrázků s plným rozlišením. To se projeví hlavně u alb, které jsou zcela naplněny obrázky (Picasa má určené maximální množství obrázků, které může obsahovat jedno album. Tento limit je 500 obrázků v jednom albu).

Stahování obrázků již není potřeba provádět pomocí skriptů napsaných v jazyce PHP. Obrázky se nenachází na doméně `picasaweb.google.com`. Jsou umístěny na doméně

googleusercontent.com, kde je v kořenovém adresáři umístěn soubor crossdomain.xml a ten povoluje přístup z jakékoliv jiné domény.

6 Závěr

Cílem bakalářské práce bylo vytvořit webovou aplikaci, která bude založena na architektuře mashup. Tedy aplikace pracující s daty jiných služeb na internetu přes jejich rozhraní tzv. API. V této práci se jednalo konkrétně o využití služeb poskytujících obrázky a fotografie svých uživatelů.

Vyvinul jsem webovou aplikaci technologií Silverlight, která umožňuje uživateli získat obrázky a fotografie z různých zdrojů, jak z lokálních, tak i ze zdrojů na internetu. Získané obrázky je možné modifikovat pomocí efektů a poté vkládat do vytvořeného schématu komiksu. Také je možné vkládat texty k jednotlivým obrázkům pro dokreslení příběhu komiksu. Vytvořený komiks si poté může uživatel uložit. Cíl této práce byl splněn. Aplikace je veřejně přístupná na adrese `komix.area59.eu`. A webová prezentace práce je na adrese `bp.area59.eu`.

Vytváření této bakalářské práce pro mě mělo velký přínos. V praxi jsem mohl aplikovat znalosti o webu a webových technologiích, které jsem během studia získal. Za největší přínos považuji zlepšení ve znalostech technologie Silverlight a programování v C#. Dříve jsem již pracoval s technologií Adobe Flash, která je v podstatě konkurencí Silverlightu. Takže jsem rád, že se v dané oblasti vývoje webových aplikací lépe orientuji.

6.1 Možná budoucí rozšíření

Do budoucna je možné rozšířit zdroje obrázků o další služby na internetu, které poskytují potřebné API. Přidat další možnosti editace obrázku do komiksu, např. přetočení obrázku nebo jeho oříznutí. Dále bych uvažoval o tom, že by si uživatelé mohli své výtvořky ukládat na server a sdílet mezi sebou a to pomocí vnitřního systému aplikace nebo sdílením na sociálních sítích. Uložené komiksy by ostatní uživatelé mohly hodnotit známkami. Uživatel by mohl nastavit přístupová práva ke komiksům uloženým na serveru (např. soukromý, veřejný).

Literatura

- [1] McCLOUD S.: *Understanding Comics – komiksový průvodce historií a teorií komixu*, Crew Comics magazín č.21/2002, ISBN 1211-6548.
- [2] SHELLY, Gary B.; FRYDENBERG, Mark. *Web 2.0 : Concepts and Applications*. Boston : Course Technology, 2010. 296 s. ISBN 978-1-4390-4802-3.
- [3] CLARKIN, Larry; HOLMES, Josh. Enterprise Mashups. *MSDN Architecture Journal* [online]. 2009, 13, [cit. 2011-04-29]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/architecture/bb906060.aspx>>.
- [4] HINCHLIFFE, Dion. *ZDNet* [online]. 2007 [cit. 2011-05-02]. Mashups: The next major new software development model?. Dostupné z WWW: <<http://www.zdnet.com/blog/hinchcliffe/mashups-the-next-major-new-software-development-model/106>>.
- [5] PEENIKAL, Sunilkumar. *Mashuos and the Enterprise* [online]. [s.l.] : Mphasis, 2009 [cit. 2011-05-05]. Dostupné z WWW: <http://www.mphasis.com/pdfs/Mashups_and_the_Enterprise.pdf>.
- [6] FIELDING, Roy . *Architectural Styles and the Design of Network-based Software Architectures* [online]. Irvine, 2000. 180 s. Dizertační práce. University of Carolina. Dostupné z WWW: <http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf>.
- [7] MALÝ, Martin. *Zdroják.cz* [online]. 2009 [cit. 2011-05-16]. REST: architektura pro webové API. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/rest-architektura-pro-webove-api/>>.
- [8] Yahoo!. *Flickr Services* [online]. 2009 [cit. 2011-05-16]. The App Garden. Dostupné z WWW: <<http://www.flickr.com/services/api/>>.
- [9] Google. *Nápověda Picasa* [online]. 2011 [cit. 2011-05-16]. Album Privacy: Authorization key. Dostupné z WWW: <<http://picasa.google.com/support/bin/answer.py?answer=48446>>.
- [10] Microsoft. *MSDN Library* [online]. 2010 [cit. 2011-05-06]. HTTP Communication and Security with Silverlight. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/cc838250\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc838250(v=vs.95).aspx)>.
- [11] LACKO, Luboslav. *Silverlight : Výukový průvodce tvorbou interaktivních aplikací*. Vydání první. Brno : Computer Press, 2010. 464 s. ISBN 978-80-251-2716-2.
- [12] WATSON, Ben. *C# 4.0 : řešení praktických programátorských úloh*. První vydání. Brno : Zoner Press, 2010. 656 s. ISBN 978-80-7413-094-6.

- [13] Microsoft. *MSDN Library* [online]. 2011 [cit. 2011-05-06]. Writing HLSL Shaders in Direct3D 9. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/bb944006\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb944006(v=VS.85).aspx)>.
- [14] KOSEK, Jiří. *PHP : tvorba interaktivních internetových aplikací*. První vydání. Praha : Grada Publishing, 2004. 492 s. ISBN 80-7169-373-1.
- [15] Microsoft. *MSDN Library* [online]. 2010 [cit. 2011-05-07]. Pixel Shader Effects. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/dd901594\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/dd901594(v=vs.95).aspx)>.
- [16] Microsoft. *MSDN Library* [online]. 2010 [cit. 2011-05-07]. ShaderEffect Class. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/system.windows.media.effects.shadereffect\(v=VS.95\).aspx](http://msdn.microsoft.com/en-us/library/system.windows.media.effects.shadereffect(v=VS.95).aspx)>.
- [17] Microsoft. *MSDN Library : Silverlight* [online]. c2011 [cit. 2011-05-06]. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/cc838158\(v=VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc838158(v=VS.95).aspx)>.
- [18] *Picnik the blog* [online]. 2010 [cit. 2011-05-08]. Picnik Acquired by Google. Dostupné z WWW: <<http://blog.picnik.com/2010/03/google-acquires-picnik/>>.

Seznam příloh

Příloha 1. CD (obsahuje zdrojové kódy aplikace, plakát)