

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Multimediální výukové prostředí pro potřebu výuky
odborných předmětů na střední škole

Jiří Honz

Bakalářská práce

2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří HONZ**
Osobní číslo: **I07620**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Multimediální výukové prostředí pro potřebu výuky odborných předmětů na střední škole**
Zadávací katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Teoretická část * Tato práce se zabývá výukovým prostředím, jehož cílem je usnadnit přístup k informacím s využitím moderních multimediálních technologií. * V úvodní části práce je nutné provést přehled problematiky a navrhnout několik možností řešení.

Implementační část * Primárním cílem práce je realizace multimediální aplikace s možnostmi: Hypertextových odkazů mezi jednotlivými kapitolami. Prohlížení obrazových příloh z jednořádkové a plošné (možnost volby) galerie miniatur podle potřeby s posuvníkem a možností volby velikosti miniatur. Zobrazování a manipulace (zvětšování a zmenšování, posun po obraze, listování dopředu a dozadu, podpora s možností kreslicího pera a zvýrazňovače) s obrazovými přílohami nejpoužívanějších formátů (BMP, GIF, JPG, PCX, PNG, TGA, TIF, WMF,?). Přehrávání video, audio a flash souborů nejpoužívanějších formátů (AVI, MPG, MOV, SWF, WMV, AIF, MP3, WAV, WMA, MP4,?). Přehrávání DVD disků a DVD z alternativních zdrojů (např. HDD) Zobrazování a manipulace s formátem PDF. Zobrazování a manipulace s formátem DWG A DXF.

* Podmínky na zpracování: Aplikace bude zpracována v souladu se zásadami typografie, teorie barev ve webovém prostředí a plně validní pro použité technologie (HTML a XHTML, CSS, JavaScript, .net, ActionScript, apod.). Aplikace musí být přenosná (portable applications) z tohoto důvodu upřednostnit produkty třetích stran, kde se nevyžaduje instalace softwaru, případně jiné možné řešení. Použití produktů třetích stran používat v rámci platných licenčních podmínek.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SUEHRING, Steve. JavaScript krok za krokem, Praha, Computer Press, 2008
2. DRIFT, Jeff, LLOYD, Ian, RUBIN, Dan. Mistrovství v CSS, Pokročilé techniky pro webové designéry a vývojáře, Praha, Computer Press, 2007
3. HAUSER, Marianne, HAUSER, Tobias, WENZ, Christian. HTML a CSS, Velká kniha řešení, Praha, Computer Press, 2006
4. POWELL, Thomas A. Web design, Kompletní průvodce, Praha, Computer Press, 2004
5. ŽIŽKA, Josef. . Webový prohlížeč přednášek [online]. Diplomová práce, Fakulta informačních technologií VUT Brno : 2009. Dostupný z WWW: <<http://www.fit.vutbr.cz/study/DP/DP.php?id=8660&y=2008>>
6. Přenosné aplikace (portable applications) [online]. Dostupný z WWW: <<http://portable.own.cz/>>
7. Validátor XHTML a HTML [online]. Dostupný z WWW: <<http://validator.w3.cz/>>
8. Dílna dobrého stylu [online]. Dostupný z WWW: <<http://wellstyled.com/>>
9. Petr Staníček, webdesigner [online]. Dostupný z WWW: <<http://www.pixy.cz/>>
10. Jak psát web, návod na html stránky [online]. Dostupný z WWW: <<http://www.jakpsatweb.cz/>>
11. Webdesign a vývoj aplikací [online]. Dostupný z WWW: <<http://interval.cz/>>
12. Referenční příručka jazyka ActionScript 3.0 a jeho komponent [online]. Dostupný z WWW: <http://help.adobe.com/cs_CZ/AS3LCR/Flash_10.0/>

Vedoucí bakalářské práce:

Ing. Zbyněk Kopecký

Katedra informačních technologií

Datum zadání bakalářské práce:

15. ledna 2010

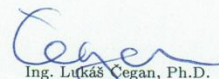
Termín odevzdání bakalářské práce:

14. května 2010



prof. Ing. Simeon Karamazov, Dr.

děkan



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 9. 5. 2010

Jiří Honz

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi nějakým způsobem pomohli, nebo při mně stáli při tvorbě bakalářské práce. Jmenovitě bych chtěl poděkovat Pavlu Jandějskovi, mému spolužákovi, který byl ochotný a dokázal poradit vždy, když jsem to potřeboval. Dále bych rád poděkoval především mé přítelkyni, která byla nejen tolerantní k mé velké zaneprázdněnosti, ale zároveň i podala pomocnou ruku kdykoliv to bylo potřeba. Samozřejmě bych tímto poděkoval i všem okolo, se kterými jsem se během uplynulého období vídal a kteří se mnou měli tu trpělivost.

Anotace

Práce se zabývá problematikou a řešením multimediální aplikace pro podporu výuky. V první části ukazuje různé možnosti řešení a návrhu. V praktické části se zabývá konkrétním návrhem a popisuje jeho celkovou tvorbu. Práce je také zaměřena na všechny použité technologie a věnuje se důležitým komponentám v prostředí Flash.

Klíčová slova

Flash, Přenositelná aplikace, Multimedia, CSS, Třída, Tween, HTML, ActionScript, JavaScript, CS4, Galerie, Časová osa, XML, URL, SWF.

Title

Multimedia learning environment for teaching technical subjects at secondary school.

Annotation

This work is dedicated to solving problem with multimedia application for education assistance. First part contains some examples and concludes of some implementations. The practical part is dedicated to one main concept and it's described creation. Another part is about all usable technologies and components in Flash.

Keywords

Flash, Portable application, Multimedia, CSS, Class, Tween, HTML, ActionScript, JavaScript, CS4, Gallery, TimeLine, XML, URL, SWF.

Obsah

| | |
|--|-----------|
| Seznam zkratk | 8 |
| Seznam obrázků | 9 |
| 1 Úvodní informace | 10 |
| 2 Řešení problematiky a různých návrhů řešení | 11 |
| 2.1 Základní rozvržení | 11 |
| 2.2 Galerie | 12 |
| 2.3 Doplnkový design..... | 13 |
| 2.4 Microsoft Silverlight | 14 |
| 2.4.1 Vývoj | 14 |
| 2.4.2 XAML | 15 |
| 2.4.3 Moonlight..... | 15 |
| 2.5 JavaFX..... | 16 |
| 2.5.1 Historie..... | 16 |
| 2.5.2 JavaFX Script | 16 |
| 2.6 Srovnání vývojových platforem..... | 16 |
| 3 Použité technologie | 17 |
| 3.1 Technologie Flash | 17 |
| 3.1.1 Vývoj a historie | 17 |
| 3.2 ActionScript..... | 18 |
| 3.2.1 Syntaxe AS3 | 19 |
| 3.3 XML..... | 20 |
| 3.3.1 Koncept | 20 |
| 3.3.2 Syntaxe..... | 21 |
| 3.4 HTML | 21 |
| 3.5 CSS..... | 21 |
| 3.6 JavaScript | 21 |
| 4 Tvorba multimediální aplikace | 22 |
| 4.1 Popis a vzhled | 22 |
| 4.2 Symboly a časová osa | 23 |
| 4.3 Rozbalovací menu | 24 |
| 4.3.1 Propojení Flashe s jazykem XML..... | 25 |

| | |
|---|-----------|
| 4.4 Komponenta FLVPlayback..... | 26 |
| 4.4.1 Syntaxe..... | 26 |
| 4.4.2 Importování..... | 27 |
| 4.5 Tweener..... | 28 |
| 4.5.1 Animace..... | 28 |
| 4.6 Odkazy..... | 30 |
| 5 Závěr..... | 31 |
| Zdroje | 32 |
| Příloha A – Adresářová struktura | 33 |

Seznam zkratek

| | |
|-------|--|
| HTML | HyperText Markup Language |
| XHTML | Extensible HyperText Markup Language |
| CSS | Cascading Style Sheets |
| AS3 | ActionScript 3 |
| JS | JavaScript |
| XML | Extensible Markup Language |
| PDF | Personal Digital Assistant |
| URL | Uniform Resource Locator |
| SWF | Small Web Format |
| RIA | Rich Internet Application |
| DOM | Document Object Model |
| WPF | Windows Presentation Foundation |
| XAML | Extensible Application Markup Language |

Seznam obrázků

| | |
|--|----|
| Obrázek 1 – Horizontální menu..... | 11 |
| Obrázek 2 – Vertikální menu..... | 12 |
| Obrázek 3 – Horizontální galerie..... | 12 |
| Obrázek 4 – Vertikální galerie..... | 13 |
| Obrázek 5 – Galerie do mřížky..... | 13 |
| Obrázek 6 – Platforma Silverlight (zdroj: Microsoft Silverlight) | 14 |
| Obrázek 7 – Jak šla doba s Flashem (zdroj: www.appleinsider.com)..... | 18 |
| Obrázek 8 – Syntaxe XML (zdroj: kirupa.com)..... | 21 |
| Obrázek 9 – Náhled aplikace..... | 22 |
| Obrázek 10 – Časová osa..... | 23 |
| Obrázek 11 – Náhled knihovny | 24 |
| Obrázek 12 – Komponenta FLVPlayback..... | 26 |
| Obrázek 13 – Import tříd | 29 |

1 Úvodní informace

V dnešní době, kdy jsou informace stále důležitější, je přístup k nim snad ještě důležitější. Mnoho nástrojů se zabývá přístupem k informacím ať už prostřednictvím televizím, rádiím, či počítačům. Právě počítače a především internet si v dnešní moderní době ukrojily veliký kus pro sebe. I prezentace a podpora distribuce informací se tak vyvíjí. Již déle nestačí pouhý psaný text, ale především je kladen důraz na multimedia, způsobu jakým se prezentují, a do jaké míry jsou interaktivní. Pro podporu výuky jakožto výukového nástroje, toto platí dvojnásob. Zajistit nástroj, který kombinuje všechny moderní techniky v přehrávání videí, zobrazování a úpravy obrázků, práci se soubory a to všechno v přenositelné formě, která by nebyla náročná na práci, to je úkolem a smyslem právě této bakalářské práce.

V první části se pokusím nastínit danou problematiku a její možnosti řešení. Velká část je věnována možným způsobům realizace. Několikrát se budu zmiňovat, že problémy spojené s multimediální aplikací jsou dost specifické a jako takové mají na jednu stranu mnoho způsobů řešení, na tu druhou se zase mohou zdát značně omezené. Budu se věnovat neméně důležité grafické části, na které v dnešní moderní době opravdu záleží a zároveň rozeberu podporu multimédií z pohledu různého softwaru, k tomu určeného.

V praktické části pak volně navážu na informace z části teoretické. Bude se v ní zabývat už konkrétně mnou navrženou aplikací a popisovat její důležité dílčí úseky.

V závěru se budu věnovat celkovému shrnutí a zhodnocení mé práce. Zamyslím se nad potenciálním rozšířením do budoucna a uvedu, co pro mě tato práce znamenala, a jak jsem se na ní obohatil.

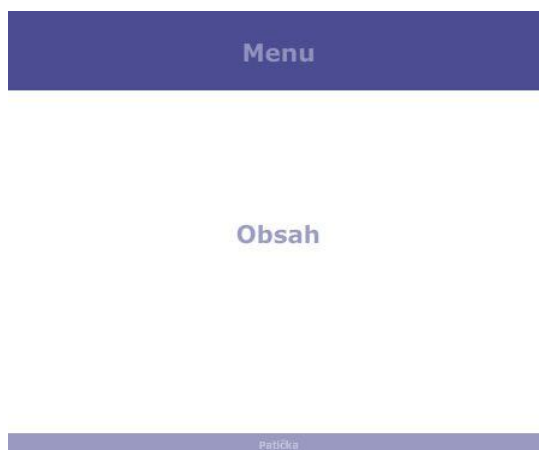
2 Řešení problematiky a různých návrhů řešení

Řešení multimediální aplikace je značně specifický problém. Návrhů řešení může být několik. Nejsou kladeny žádné meze, co se navržení, designu a celkového uspořádání týče. Avšak způsoby provedení jsou omezené. Já jsem si vybral program Adobe Flash CS4, který se spolu s XML, HTML, CSS a JavaScriptem stává velice mocným a komplexním nástrojem. Podle mého asi nejvhodnější řešení. Jeho podpora je sice v České Republice poněkud menší, ale v zahraničí se mu dostává náležitě pozornosti, a tak se studijní podpory a materiály daly sehnat. Podpora v tomto případě tak nakonec byla vcelku velká. Jak už bylo zmíněno, moc jiných možností řešení pro tento problém nakonec není, přesto se však pokusím o některém zmínit.

2.1 Základní rozvržení

Základem všeho je správné rozvržení. Především se to týká prvotnímu promyšlení o celé funkčnosti, použitelnosti a neméně důležité součásti, celkovému stylu a vzhledu. Vše se muselo připravit tak, aby se mohla realizovat například obrazová galerie, která může být řešena nepřeborným množstvím možností. Např. galerie v podobě takzvané Slideshow¹, nebo galerie obrázků řešená horizontálně jednořádkově, popřípadě vertikálně. Ale existují i další řešení, třeba navrhnutí srovnání a seřazení do mřížky apod.

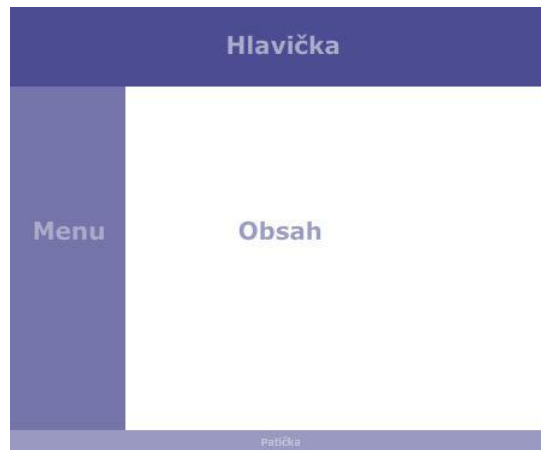
Vrátím se ale na začátek. Pro něj je nejdůležitější vůbec navrhnout nějaký vkusný design. Je potřeba osvojit si alespoň nějaké základy Webdesignu a čerpat z nich. Základ všeho je pěkný a použitelný layout². Znovu je zde plno možností jak toho docílit. Je nutné si zezáčátku stanovit omezenou paletu barev a z ní čerpat. Umístění položek menu, hlaviček nebo patiček je znovu výběr z mnoha možností. Menu můžete umístit nahoru do hlavičky a nechat tak větší a širší místo na obsah, nebo můžete zvolit menu na stranách a uspořádat tak výsledné stránky jinak. Zde je možností opravdu nepřeborné množství a způsobů jak realizovat celkový vzhled mnoho. Na ukázkou dva způsoby návrhů:



Obrázek 1 – Horizontální menu

¹ Zobrazení série obrázků, především pro praktické a umělecké užití

² Grafický návrh zobrazení stránky



Obrázek 2 – Vertikální menu

Zde jsou vidět na ukázkou dvě základní rozvržení, se kterými se dá dál experimentovat a vytvořit naprosto odlišné styly. Nebo se dají stávající upravovat k obrazu svému.

2.2 Galerie

Tou další důležitou náplní práce bylo vytvoření takzvaných „galerií“ ať už pro práci s obrázky, tak i pro práci s videem či soubory. Řešení těchto galerií jsem již v předchozí kapitole započal, a proto bych to teď ještě rozšířil a ukázal i ukázky těchto galerií. Základním kamenem je tedy celkový layout, jak již je navrhnutý, je nutné ostatní prvky řídit podle něj. To platí právě i pro tyto galerie, které tedy mohou být řešeny vícero způsoby. U většiny příkladů se jedná o „vertikální³ versus horizontální⁴“. Například:



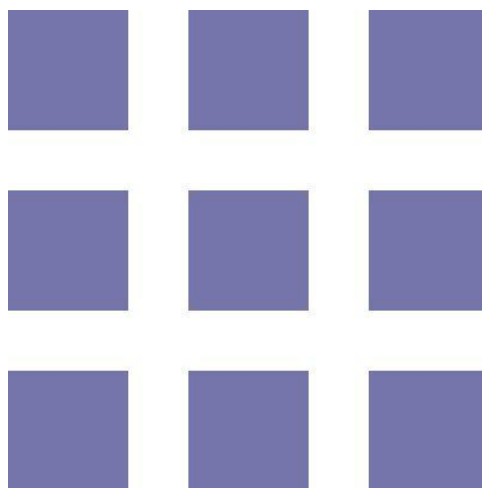
Obrázek 3 – Horizontální galerie

³ Svisle (nahoru, dolů)

⁴ Vodorovně (zprava, doleva)



Obrázek 4 – Vertikální galerie



Obrázek 5 – Galerie do mřížky

I tady se dají navrhovat netradiční řešení a realizovat je všemi možnými způsoby. Samotné takovéhle navržení ale nestačí. Musí se počítat se způsoby rozšíření a to především v dynamickém doplňování galerií a přidáváním obrázků. Takový to problém se dá pak řešit například posuvníky. Vytvořením jakékoliv lišty, která podle potřeby bude jezdit jezdcem. Nebo přepínání v seznamu za použití jednoduchých tlačítek. Především Flash je v tomto dost rozmanitý a například mnou použitý způsob je postaven na pohybu myši a posouvání se po galerii daným směrem.

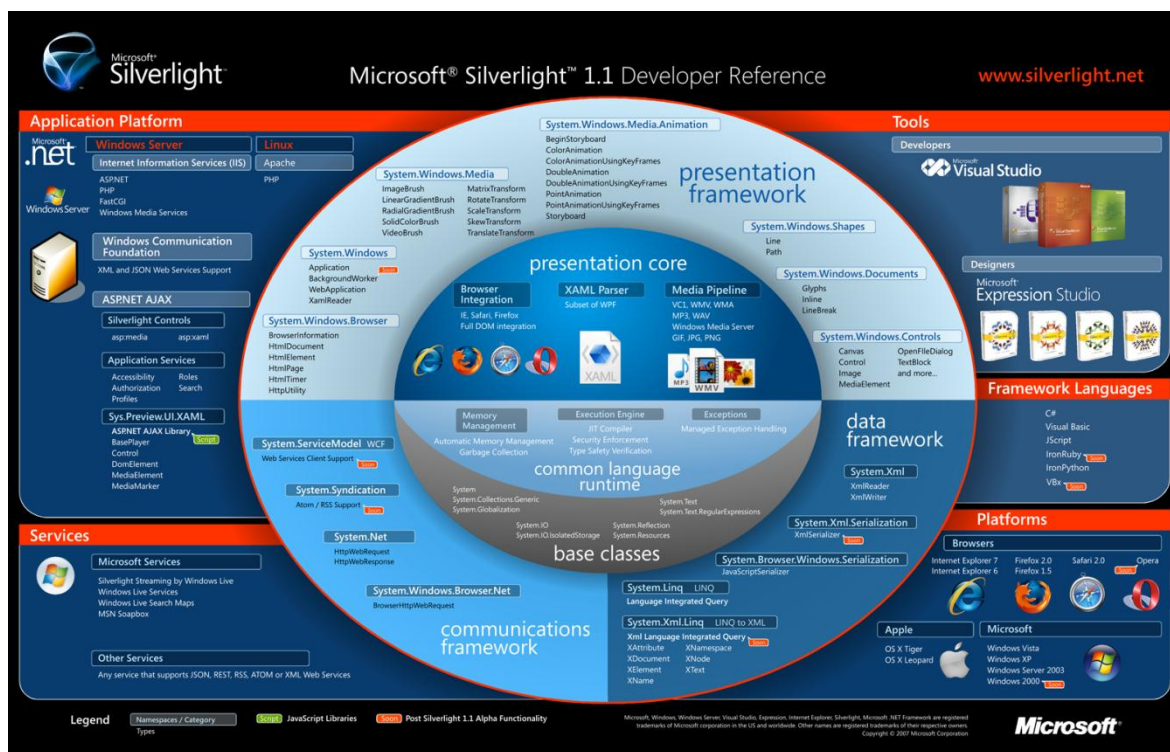
2.3 Doplnkový design

Samozřejmě se kompletní návrh nesestává jen z jádra samého. Většinou se dá upravovat spousta věcí okolo. Ať už je to dobré pro snadnější práci nebo funkčnost, tak i pro oko. Od zasouvacích lišt, přes vytvoření dalšího rozhraní, i přes další vymoženosti, která vám jsou umožněny v rámci daného programu. Například změny stínování, překrývání, animace a pohyb kurzoru atd.

2.4 Microsoft Silverlight

Přejdu od designu k postupné proveditelnosti. Já se rozhodl pro Adobe Flash CS4, jak jsem se již stihl zmínit. Především díky tomu, že mi je bližší a také že má v zahraničí celkem dobrou podporu a přeci jen jsem o něm podvědomě věděl nejvíce. Přeci jen je tu ale jeden nástroj, který se s Flashem ztotožňuje nejlépe a pro podobnou práci má zase své proveditelné možnosti. Je jím Microsoft Silverlight.

Microsoft Silverlight je moderní platforma určená k tvorbě dynamického obsahu, animací, vektorové i bitmapové grafiky, textu a ostatního obsahu podporujícího interaktivitu. Díky stažitelné komponentě (plug-in) umožní interaktivní práci s multimédií, hrami či jinými aplikacemi. Podpora je zajištěna na většině současných prohlížečích, a to samé se dá říci i o operačních systémech.



Obrázek 6 – Platforma Silverlight (zdroj: Microsoft Silverlight)

2.4.1 Vývoj

V současnosti se Silverlight nachází již ve 4. verzích. Platforma jako taková je postavena na WPF⁵ a .NET Framework 3.0⁶ a slouží především ke tvorbě webových aplikací.

První verze 1.0 vyšla v roce 2006 a jako programovací jazyk používala JavaScript. Měla podporu 2D grafiky a obrázků. Textů, animací, audia i videa, či práci se sítí. Její komunikace probíhala přes AJAX⁷.

⁵ Ang. (Windows Presentation Foundation) dříve známý jako Avalon

⁶ Soubor technologií v softwarových produktech

Druhá verze 2.0 (původně označována jako 1.1. Změna přišla v roce 2007) už jako programovací jazyk nevyužívala jen Javascript, nýbrž C# či prostředí .NET Framework. Přidala navíc velmi důležitou podporu tlačítek, všemožných ovládacích prvků a různých dalších komponent, jako například panel, přepínač, seznam atd.

Třetí verze 3.0 pak přišla ještě s daleko větším rozšířením stávajícího stavu. Byla přidána podpora 3D, objevily se zde nové ovládací prvky a došlo i na podporu SEO. Jako dalším důležitým bodem se zde ukázala taky nově zpřístupněná podpora „offline“ aplikací.

Plně funkční a plná čtvrtá verze se objevila teprve nedávno, 15 dubna 2010. Mimo dalšího rozšíření a jiných úprav je zde důležitá podpora prohlížeče Google Chrome, který se stává čím dál tím víc populárnější, a další neméně důležitou informací je i podpora webové kamery a mikrofonu. Nejnovější verze je tak zatím označována jako Silverlight 4.

2.4.2 XAML

XAML je v aplikacích Silverlight používán především k definování uživatelských rozhraní. Je využíván pro zobrazení vektorové grafiky a animací. Je to jazyk společnosti Microsoft, který je založen na XML. Práce s XAML je velice jednoduchá a tak je používán již od .NET Frameworku 3.0. Dá se s ním pracovat jak klasickou úpravou textového souboru, tak například i speciálním softwarem jako je například XAMLPad editor. Dá se také říci, že platí: To co si vytvoříte v XAML, můžete alternativně vytvořit i v některých z .NET jazyků, například C#.

Jelikož je tento jazyk postavený na XML, tak i jeho zápis je velice podobný. Takový rychlý a krátký ukázkový příklad:

```
<Canvas xmlns="http://schemas.microsoft.com/winfx/2006/xaml ">
  <TextBlock>Vítej, bakalářská práce! </TextBlock>
</Canvas>
```

Krátký příklad v aplikaci Silverlight, který vypíše vše, co je obsaženo mezi elementy <TextBlock> a </TextBlock>.

2.4.3 Moonlight

Aby byla zajištěna multiplatformost, byla firmou Novell vyvinuta implementace Moonlight. Což je jakési volně šířitelné rozhraní technologie Silverlight pro Linux. Jelikož ten nebyl základním „Silverlightem“ podporován a bylo mu to dost vyčítáno. Je to také součástí projektu Mono⁸, který je veden právě firmou Novell.

⁷ Ang. (Asynchronous JavaScript and XML) technologie vývoje interaktivních webových aplikací

⁸ Projekt, jehož cílem je vytvořit sadu nástrojů pro práci v prostředí .NET a zajistit dostupnost na všech operačních systémech

2.5 JavaFX

Platformy Adobe Flash a Microsoft Silverlight se za ta léta dočkala masového rozšíření a za tu dobu si stihly upevnit svoji pozici na poli nejen webových aplikací. Jako odpověď na toto „prosazení“ přišla společnost Sun Microsystems s vlastní softwarovou platformou, JavaFX. Jedná se o jednoduchý a přívětivý nástroj pro tvorbu RIA aplikací. Tato platforma je postavena na Javě samotné a proto nejen že poskytuje interaktivitu na poli webových aplikací, ale zároveň je umožňuje stahovat a používat je jako „offline“ desktopové aplikace.

2.5.1 Historie

- Květen 2007 – První oznámení JavaFX na konferenci vývojářů
- Květen 2008 – Oznámení o uvolnění platformy do konce roku pro prohlížeče a desktopy. Zároveň sdělení budoucího plánu o uvolnění pro mobilní zařízení
- Konec července 2008 – plugin pro NetBeans 6.1
- 12. Února 2009 – JavaFX 1.1. Zpřístupněna podpora pro mobilní zařízení
- 2. Června 2009 – Na konferenci JavaOne vydána verze 1.2
- 22. Dubna 2010 – Vydána nejnovější verze 1.3

2.5.2 JavaFX Script

Specializovaný skriptovací jazyk používaný pro účely platformy JavaFX. Původně pojmenován F3⁹. Jedná se o statický deklarativní skriptovací jazyk, který své kořeny čerpá z jazyka Java. Vytvořené soubory mají koncovku .fx. Jeho architektura je taková, že kromě základního skriptovacího jazyka kombinuje i množství vývojových nástrojů, spousty grafických knihoven, audio knihoven atd.

2.6 Srovnání vývojových platforem

Srovnání prostředí, ve kterých jde požadovanou multimediální aplikaci navrhnout, není věc jednoduchá. I přes pár možných typů implementace, jich přeci jen není tolik, aby šlo vše správně provést. Především v oblasti multimédií se jedná jen o Adobe Flash, Microsoft Silverlight, či JavuFX od Sun Microsystems. Pravdivé tvrzení je to, že každá z uvedených technologií, si pojede právě na těch svých, které jim společnosti podporují. U Silverlightu to bude především prostředí .NET a systémy firmy Microsoft obecně, u JavyFX zase řešení firmy Sun Microsystems a jazyka Java. Adobe Flash bude stavět především na systémech společnosti Adobe, i přes to ale z této trojice vychází z mého pohledu nejlépe. Vybral jsem si ho tedy nejen pro to, že je mi bližší a že má vyšší podporu ve světě, co se materiálu a informací týče, ale také proto, že technologie Flash je vyvíjena už několikátým rokem a pořád si udržuje obrovskou podporu. Vždyť právě v této době vychází již další verze, Adobe Flash CS5.

⁹ Ang.(Form, Follows, Function)

3 Použité technologie

Z důvodů provedení a dodržení zásad o funkčnosti aplikace byl výběr použitých technologií značně omezen. Budeme vycházet z předchozího shrnutí. Jelikož aplikace má sloužit především jako přenositelná, bez přístupu na internet, musela se tak zajistit její bezproblémová přístupnost. Zároveň se však musela dodržet jistá webová podoba. Pro to se zvolil program Adobe Flash CS4 s podporou skriptovacího jazyka AS3. Jako alternativa byl dodržen vzhled stránky HTML na stylován pomocí CSS a upraven do podoby své právě pomocí souborů SWF, vycházejících z Flashe, a JavaScriptu. Pro podporu dynamičnosti a lepší správy souborů jsem zvolil XML, které bylo do prostředí Flashe připojeno.

3.1 Technologie Flash

Flash je vývojové prostředí pod hlavičkou společnosti Adobe. Nejen označován jako grafický vektorový program. Používá se především ke tvorbě animací, her, prezentací nebo různých interaktivních aplikací (zejména v oblasti internetu). Dnešní době je využíván především ke tvorbě takzvaných RIA¹⁰. Flash má dále importovaný vlastní programovací jazyk, ActionScript, z něhož se stal během let vývoje mocný objektově orientovaný nástroj, který možnosti Flashe jako takového, značně rozšiřuje.

3.1.1 Vývoj a historie

Získán firmou Macromedia, Flash byl uveden roku 1996 a nyní, jak už bylo zmíněno, je dál vyvíjen a distribuován firmou Adobe. Předchůdce programu Flash, byl SmartSketch¹¹, aplikace která běžela na operačním systému PenPoint¹², což byl systém pro grafické tablety. Postupem času byl SmartSketch znovu vydán jako FutureSplash, vektorově orientovaná webová animace, která v té době soutěžila s Macromedia Shockwave. V roce 1995 byl SmartSketch předělán na snímkovou animaci (dnešní podoba) a v již zmíněném roce 1996 byl získán firmou Macromedia, která z něj „udělala“ Flash. Což je vlastně složenina z názvů „Future“ a „Splash“.

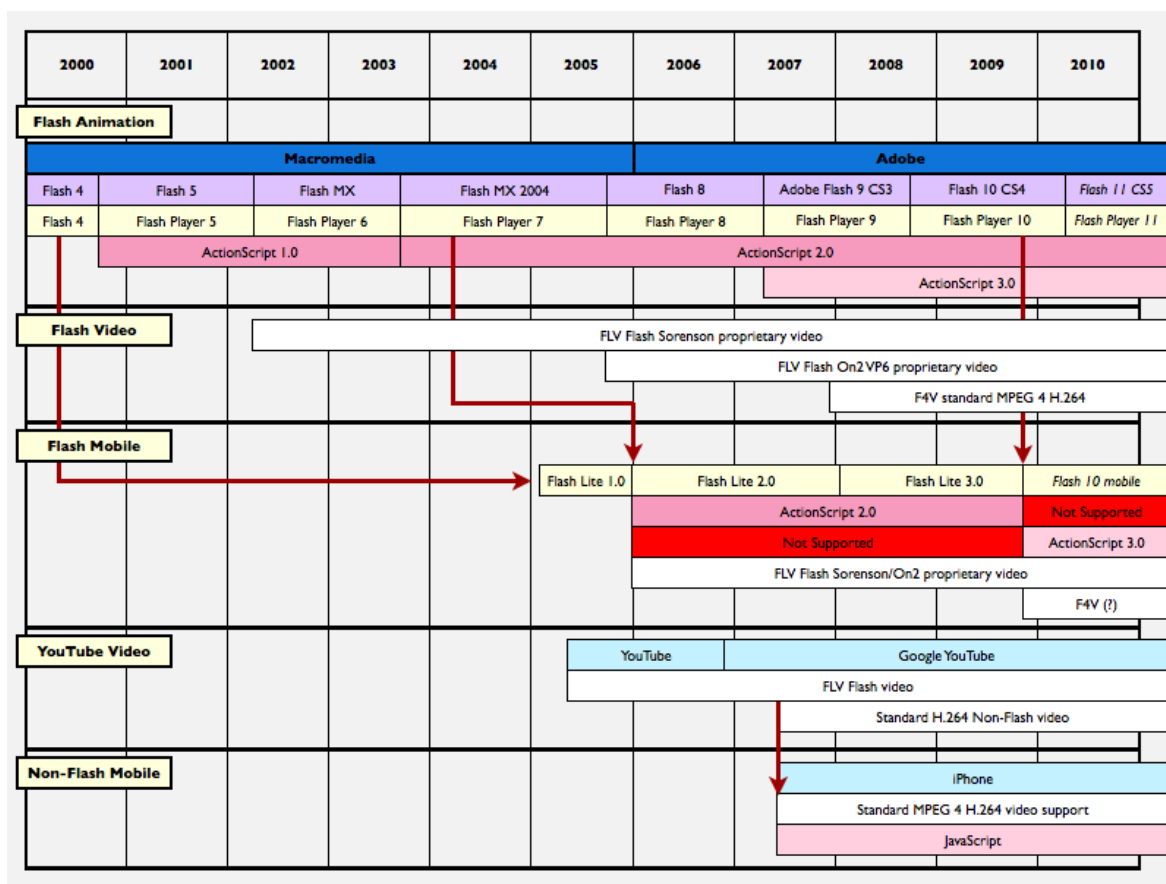
S Flashem postupem času vznikaly stále nové a nové verze. Ať už u Macromedie číselná označení, nebo potom u firmy Adobe přídavek CS. Nebyly to ale jen verze Flashe samotné. Spolu s ním se stále vyvíjí i přehrávač Flash Player, který je v dnešní době již na nějaké 10. verzi, ale vyvíjí se právě třeba již zmiňovaný ActionScript, který z nějaké své první verze ušel obrovský krok a dnes se již ve své 3. verzi může pyšnit daleko dokonalejším zpracováním, než v předchozím raném stádiu. S přibývajícím dobou se navíc Flash dostával stále k širší a širší veřejnosti a jeho podpora je v dnešní době skoro až 100%.

¹⁰ RIA (z ang. Rich Internet Application) bohaté internetové aplikace

¹¹ Kreslicí aplikace pro tužkové počítače

¹² Vyvíjeném Jonathanem Gayem

Postup a částečné zobrazení vývoje je zobrazeno na následujícím obrázku:



Obrázek 7 – Jak šla doba s Flashem (zdroj: www.appleinsider.com)

3.2 ActionScript

Je skriptovací jazyk založený na ECMAScript¹³. Je využíván především ve webových aplikacích s použitím softwaru Adobe Flash Player. V programu flash slouží k práci s interaktivitou, tvorby intuitivních prostředí, rozpohybování animací, k tvorbě stylových prezentací a grafiky, prakticky může ovládat spoustu věcí. Využívá se ale například i v databázových aplikacích jako je Alpha Five¹⁴, nebo v základech robotiky s Make Controller Kitem¹⁵.

Jak již bylo zmíněno, i jazyk ActionScript se za ta léta vyvíjel. Zezačátku byl uváděn pouze jako lepší podpora pro tvorbu animací. Teprve s nástupem verze 3 byl tento jazyk převeden do podoby rozsáhlého objektově-orientovaného prostředí, které si dokáže poradit prakticky s čímkoliv. Tvorbou tlačítek, vzhledu, kompletních webů, interaktivních aplikací a mnoho dalšího. V poslední době je jako největší krok k tvorbě moderních RIA projektů propojení Flashe a XML, které je dále využito v rozšiřujícím prostředí Adobe Flex.

¹³ Rodina skriptovacích jazyků na webu, ke které patří třeba i JavaScript

¹⁴ Především pro webové aplikace AJAX

¹⁵ Open Source Hardware

3.2.1 Syntaxe AS3

Již mnohokrát bylo zmíněno, jak si ActionScript verze 3 zakládá na objektově-orientovaném programování všeobecně. Oproti předchozím verzím je zde kladen důraz především na třídy a balíčky. Největší důraz a značné zdokonalení přišlo především v systému DOM event handling¹⁶. Kdy pojmenovaná instance reaguje na vnější událost způsobenou například kliknutím myši, vypršením čas. limitu apod.

Například:

```
Tlacitko_btn.addEventListener(MouseEvent.CLICK, MenuKliknuto);
```

Zde vidíme přiřazení k pojmenované instanci tlačítka „Tlacitko_btn“. K této instanci se váže událost „CLICK“, která reaguje na stisknutí myši nad daným tlačítkem. Celé je to pojmenované „MenuKliknuto“ což nám tvoří základ pro funkci s tímto názvem, kde musíme říci, co má daná událost vlastně provést. Například takto:

```
Function MenuKliknuto(event:MouseEvent):void  
{  
    gotoAndStop(10);  
}
```

Což je jednoduchá definice funkce, která nám říká, že po kliknutí na naše tlačítko, se přesuneme na snímek s číslem 10.

Takto tedy vypadá řízení událostí. Ovšem dalším stěžejním bodem je všeobecná syntaxe jazyka, která se skládá z objektů a tříd, které se dají většinou rozkouskovat a vytvořit tak z nich samo-spustitelné balíčky.

Například:

```
var pozdrav:TextField = new TextField();  
pozdrav.text = "Ahoj Světe";  
this.addChild(pozdrav);
```

Tady je k vidění základní deklarace proměnné „pozdrav“, která je přiřazena k typu textového pole a po nabytí hodnoty „Ahoj Světe“ je následně zase předána zpět. Důležitý je také pak import jednotlivých tříd, které flash nabízí (v současnosti se na internetu pohybuje mnoho volně šiřitelných komponent a doplňků, že si je můžete do svých projektů stahovat a pracovat s nimi. O těch ale až později) Například teda pro správné fungování našeho textového pole, musíme naimportovat třídu takto:

```
import flash.text.TextField;
```

Spousty doplňkových věcí se tak musí tímto způsobem naimportovat, protože bez nich by funkčnost nebyla zaručena. Pokud si to tedy celé shrneme do nějakého funkčního celku, mohlo by to pak například vypadat třeba takto:

¹⁶ Programování založené na řízení událostí (především ve webovém prostředí)

```

package com.example
{
    import flash.text.TextField;
    import flash.display.Sprite;

    public class Privitani extends Sprite
    {
        public function Privitani()
        {
            var pozdrav:TextField = new TextField();
            pozdrav.text = "Hello World";
            addChild(pozdrav);
        }
    }
}

```

Toto je finální podoba této malé ukázky. Celá je zabalená do jednoho balíku, který v sobě zahrnuje všechny potřebné třídy k správnému spuštění programu. Používání balíčků je jeden ze způsobů použití ale je možno je i vynechat. Syntaxe programu tak může být jiná. Na začátku se neimportují všechny potřebné prvky, pak přijdou na řadu globální proměnné a zbytek kódu je dále do funkcí a jejich dalších pod-funcí.

3.3 XML

Je rozšířený značkovací jazyk, který je definován specifikací XML verze 1.0. Jeho hlavní cíl je zajistit použitelnost a především zjednodušenou nejen v oblasti internetu. V současnosti je mnoho programových prostředí, které úzce pracují, nebo jsou přímo postaveny na XML. Využívá se tedy především pro výměnu dat mezi aplikacemi nebo publikování obsahem určených dokumentů.

3.3.1 Koncept

Celkový koncept je založen především na specifikaci XML. Může se zde objevit téměř každý textový znak, který je v Unicode¹⁷. V XML rozeznáváme takzvané elementy, což jsou závorky <>, z nichž každá musí být na své straně a element tak musí být uzavřen.

Například:

```
<Pozdrav>Ahoj</Pozdrav>
```

Tady si můžeme všimnout, že pravá ukončovací závorka musí obsahovat ještě lomítko, která představuje ukončení daného elementu. Pro XML platí, že tu musí být jeden kořenový element. Celá struktura tak pak vypadá jako jeden strom, který je odvozen od kořenové položky. Pak je důležitý také to, aby si každý zachoval své místo a elementy se zbytečně nepřeskakovaly a dodržovaly správné ukončení. K elementům je pak možno dodávat a dopisovat možné atributy, které se zapisují buď do jednoduchých, nebo do dvojítek uvozovek. Například takto:

```
<Pozdrav navez="Ahoj" vyska="100" sirka="100" ></Pozdrav>
```

¹⁷ Prezentuje práci s textem

3.3.2 Syntaxe

Celková syntaxe, která se může skládat z nepřeberného množství elementů, atributů a všeho dalšího, tak pak může vypadat třeba nějak takto:

```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```

Obrázek 8 – Syntaxe XML (zdroj: kirupa.com)

3.4 HTML

Z anglického HyperText Markup Language je překládán jako značkovací „jazyk pro hypertext“. Slouží především pro publikování dokumentů na internetu a k tvorbě webových stránek. Ve své práci ho spolu s CSS používám k alternativnímu zobrazení výsledné aplikace s podporou přepínání hypertextových odkazů.

3.5 CSS

Anglicky Cascading Style Sheets. Je to jazyk, který slouží pro zobrazování především webového obsahu a stránek napsaných v jazyce HTML, XHTML nebo XML. Jazyk nabízí obrovské množství formátování, především textu a co se grafické úpravy týče. Byl navržen organizací W3C¹⁸ a prozatím byly vydány dvě jeho specifikace, CSS1 a CSS2. Na verzi CSS3 se ještě pracuje.

3.6 JavaScript

Multiplatformní a objektově orientovaný jazyk. Využíván především pro webové aplikace a internetové stránky za účelem interaktivity a ovládání různých dynamických prvků GUI¹⁹. Používám ho jako doplněk k alternativnímu zobrazení v HTML verzi.

¹⁸ Mezinárodní konsorcium pro vývoj webových standardů

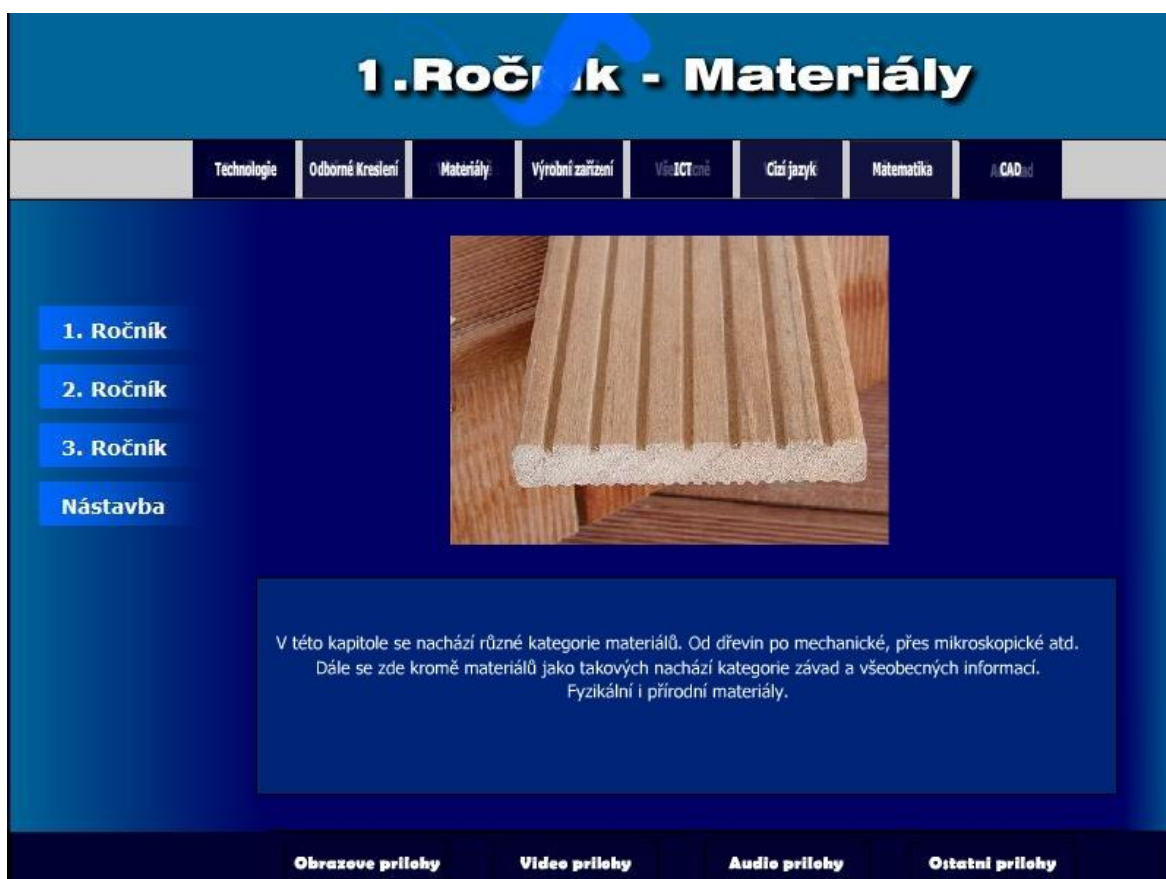
¹⁹ Grafické uživatelské rozhraní

4 Tvorba multimediální aplikace

Jak už jsem popsal výše, aplikace byla kompletně vytvořena za použití programu Adobe Flash CS4 a skriptovacího jazyku ActionScript 3. Výsledný formát je navržen jak do podoby samostatných flashových (flash player) souborů (SWF), tak alternativně i do HTML. Celá struktura funguje tak, že aplikace je rozvržena do samostatných stránek podle kapitol. Každé z těchto kapitol náleží jeden soubor .fla, který je tak nezávisle upravitelný a oddělený od ostatních. Avšak galerie, které se dají rozdělit tak, že každá z nich obsahuje samostatný ActionScript soubor (soubory typu .as), jsou vždy v jednom příslušném .fla souboru společně.

4.1 Popis a vzhled

Na následujícím obrázku se nachází ukázka aplikace, konkrétně sejmutá z kapitoly prvního ročníku a tématu broušení:



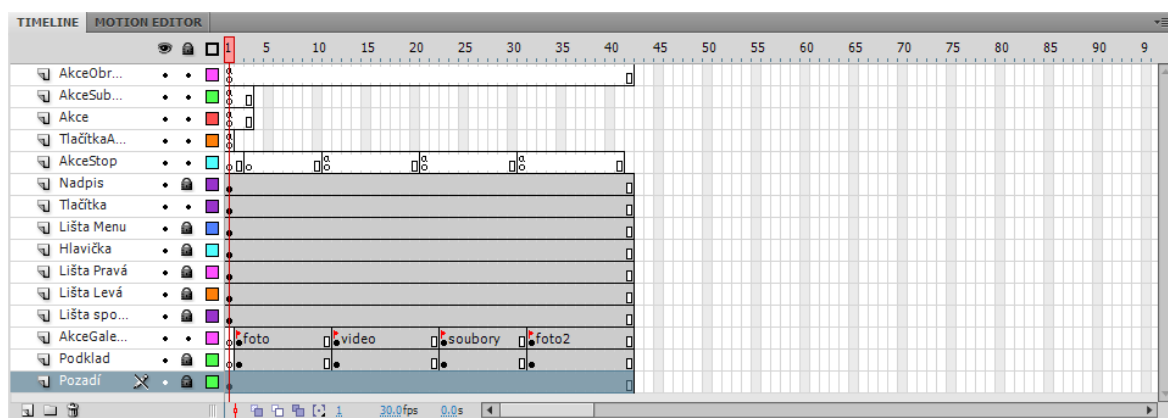
Obrázek 9 – Náhled aplikace

Na náhledu výše je možné si všimnout hned několika prvků. Na úvod bych rád poznamenal, že aplikace v sobě má určitý počet animací, které na náhledu nejsou vidět, nebo nejsou v pohybu (viz modrý „had“ na horním nadpise). Na tomto obrázku se nacházíme v sekci prvního ročníku, v kategorii Technologie a tématu broušení. Celé horní menu vyjíždí a odkrývá další pod-menu teprve po najetí myší. Na levé straně je základní

menu pro výběr ročníku a v patičce se nachází rozcestník mezi galeriemi. Všechna tyto tlačítka jsou plně interaktivní a je možné mezi nimi přepínat a vybírat si. Na snímku je zobrazená ukázková foto galerie, která je řazená do horizontální jednořádkové polohy. Pohybem myši do stran nám jezdí výběr všech ikoněk obrázků a je tu tak možno vybrat si pokaždé, bez ohledu na počet. (galerie se automaticky a dynamicky přizpůsobuje velikosti počtu obrázků). Po kliknutí na vybraný obrázek se tento obrázek zvětší a vy tam vidíte jeho plný náhled. Do něj pak můžete kreslit libovolnou barvou a vyznačovat podle libosti. Na levé straně obsahu se nachází paleta barev a zároveň i tlačítko pro vymazání všeho, co jste nakreslili.

4.2 Symboly a časová osa

Rozvržení v programu Adobe Flash představuje především sled časových os a vrstev. Dále pak seznam všemožných komponent a vlastních vytvořených symbolů. (tlačítka, filmový klip, čistá grafika) Proto jsem se to celé snažil navrhnout tak, jak mi Flash dovolil a jak se to podle mého zdálo nejlepší. Ukázka časových os a knihovny²⁰ :



Obrázek 10 – Časová osa

Na obrázku je znázorněno mé rozvržení hlavní časové osy. Toto se týká především hlavní stránky. Samozřejmě uvnitř jiných symbolů se nachází jiné, pro ně unikátní nastavení. Každý symbol galerie tak například má svůj snímek „akce“ s kódem jazyka ActionScript, který se o vše stará uvnitř klipu a nezasahuje tak do hlavního obsahu. Toto řešení je občas kritizováno, ale já ho zvolil jako nejvhodnější, protože mi to usnadnilo přehlednost a vyřešilo pár neduhů, které se především v programování a uskupení snímků s kódem jazyka ActionScript, vyskytly.

Skoro všechny grafické prvky v hlavních časových osách jsou zamknuté proto, aby se náhodou pozdější manipulací nepoškodilo jejich umístění. Manipuluje se tedy jen se snímky, které nějakým způsobem zajišťují interaktivitu a mají dopad na funkčnost. Všechny prvky jsou navíc tvořeny jako symboly a jsou uchovávány v knihovně.

²⁰ Knihovna ve Flashi je místo, kde se uchovávají všechny vaše importované, nebo vytvořené symboly

Ukázka části mé vytvořené knihovny:



| Name | Linkage | Use Count | Date Modified | Type |
|------------------|---------------------------------|-----------|--------------------|--------------|
| btn | | 0 | 13.4.2010 12:24:48 | Button |
| button | | 1 | 26.4.2010 21:05:43 | Movie Clip |
| button Galerie | | 1 | 26.4.2010 21:05:53 | Movie Clip |
| button Soubory | | 1 | 26.4.2010 21:06:21 | Movie Clip |
| button Video | | 1 | 26.4.2010 21:06:35 | Movie Clip |
| button Vymazat | | 1 | 26.4.2010 21:14:30 | Movie Clip |
| ColorPicker | Export: fl.controls.ColorPicker | 1 | 2.7.2008 18:34:37 | Component |
| Component Assets | | | | Folder |
| FLVPlayback | Export: fl.video.FLVPlayback | 0 | 9.7.2008 16:44:58 | Compiled ... |
| FotoGalerie | | 1 | 8.5.2010 14:47:00 | Movie Clip |
| FotoGalerie2 | | 1 | 8.5.2010 14:47:46 | Movie Clip |
| Hit Area | | 5 | 26.4.2010 20:19:20 | Movie Clip |
| Hlavička | | 1 | 27.4.2010 1:33:51 | Movie Clip |
| Hlavní text | | 1 | 27.4.2010 2:14:58 | Movie Clip |
| Lišta levá | | 2 | 27.4.2010 12:36:27 | Graphic |
| Lišta Menu | | 1 | 27.4.2010 1:32:33 | Movie Clip |
| Lišta Pravá | | 1 | 27.4.2010 12:38:23 | Graphic |
| Lišta spodní | | 1 | 26.4.2010 17:45:21 | Graphic |
| Menu Rectangle | Export: MenuRectangle | 120 | 26.4.2010 20:19:16 | Movie Clip |
| menuItem | Export: MenuItem | 0 | 27.4.2010 1:49:14 | Movie Clip |
| My Circle | Export: MyCircle | 1 | 27.4.2010 1:29:16 | Movie Clip |
| Nadpis | | 1 | 27.4.2010 2:06:22 | Movie Clip |
| Obsah foto | | 4 | 27.4.2010 0:53:14 | Movie Clip |
| Obsah panel | | 1 | 27.4.2010 1:11:01 | Movie Clip |
| Obsah video | | 2 | 27.4.2010 1:02:50 | Movie Clip |
| Panel | | 1 | 26.4.2010 21:24:42 | Movie Clip |
| ProgressBar | Export: fl.controls.ProgressBar | 0 | 2.7.2008 18:34:38 | Component |
| SmenuItem | Export: SMenuItem | 0 | 27.4.2010 1:58:32 | Movie Clip |
| SouboryGalerie | | 1 | 26.4.2010 15:21:19 | Movie Clip |
| Symbol 59 | | 0 | 23.4.2010 20:56:59 | Graphic |

Obrázek 11 – Náhled knihovny

Zmínil jsem se o tom, že knihovna je vlastně taková úschovna symbolů, která je má všechny uložené pro snadnou úpravu a opakovatelnou použitelnost. V knihovně můžeme dále vidět i všechny použité komponenty z našeho projektu. (Například komponenta pro přehrávání vide, FLVPlayback) Můžeme i vyčíst, kdy byl daný prvek naposledy upraven, jakého je typu a kolikrát byl použit. Mimo jiné je zde taky takzvaná položka „Linkage“. Ta slouží především pro prvky, které jsou napojeny a úzce spolupracují s jazykem ActionScript. Takové prvky jsou z knihovny přiřazeny přímo do běhu programu a práce s nimi je v tomto případně nutná.

4.3 Rozbalovací menu

Horní rozbalovací menu je kompletně vytvořeno v prostředí Adobe Flash s propojením na jazyk XML. Po grafické stránce je navrženo tak, aby ladilo do designu celé aplikace a aby zároveň nerušilo a nekazilo výsledný dojem.

4.3.1 Propojení Flashe s jazykem XML

Jak jsem již zmínil, menu je především pro lepší dynamiku a usnadnění, postaveno na základě jazyka XML. Celá jeho struktura je zde popsána a je následně napojena na jazyk ActionScript.

```
<menus>
<menu id="Technologie">

<item>Řezání</item>
<item>Vrtání</item>
<item>Dlabání</item>
<item link="MultimediaLearningSubmenu1RtechBrou.html">Broušení</item>

</menu>
</menus>
```

Tady je například ukázána jedna položka menu, konkrétně s názvem „Technologie“. Rozbalování a menu jako taková obstarává jazyk ActionScript, zde se však definují všechny podstatné atributy. Například počet a názvy položek v pod-menu a také adresy, na které jednotlivé položky ukazují. V tomto případě je to atribut link u položky Broušení.

```
var xmlPath:String = "1RocnikXML.xml";
var xml:XML;
var loader = new URLLoader();
loader.load(new URLRequest(xmlPath));
loader.addEventListener(Event.COMPLETE, xmlLoaded);
```

Tato část v jazyce ActionScript definuje cestu k souboru s příponou .xml a následně načítá funkci „xmlLoaded“, která se stará o procházení prvků a jejich načtení. Důležitá je však tato část, kde říkáme jaký typ a odkud se má vůbec nahrávat. Pokud bychom se pak podívali na funkci samotnou, tak je vytvořená zhruba takto:

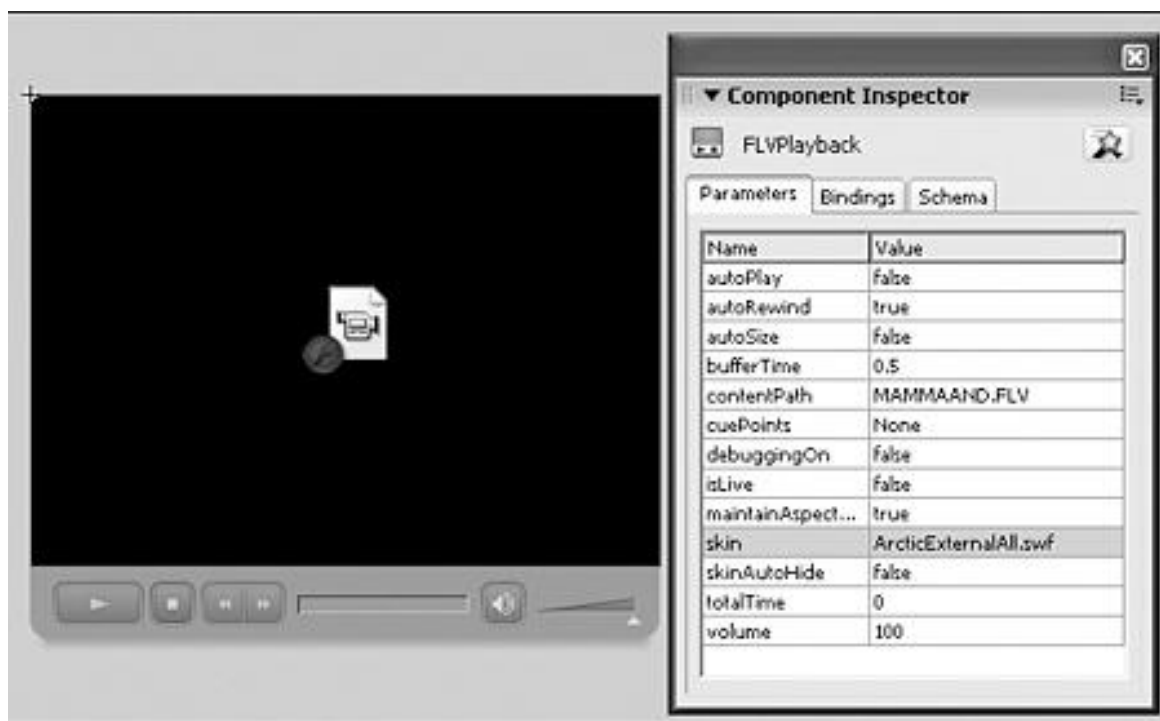
```
function xmlLoaded(e:Event):void {
    if ((e.target as URLLoader) != null ) {
        xml = new XML(loader.data);
        xml.ignoreWhitespace = true;
        createMenus();
    }
}
```

Zde je ukázka zdrojového kódu mé vytvořené funkce. Podmínka zajistí, že nebudeme pracovat s nulovým „loaderem“ a předejdeme tak dalším problémům. Nahraná data následně uložíme do proměnné „xml“ a nakonec zavoláme funkci createMenus(), která sama o sobě tvoří hlavní kořenovou strukturu menu a definuje v tomto případě další rozměry a pozice.

Za pomoci jazyku XML není tvořeno pouze menu. Na jeho základě staví prakticky celá aplikace, protože se tam uchovávají informace o obrázcích, souborech, videích a plno dalšího. V případě například galerie, se v XML definují i její rozměry a pozice, které jsou nataženy do Flashe a v něm následně upravovány.

4.4 Komponenta FLVPlayback

Ve své práci pro spuštění a manipulaci videa jsem použil právě komponentu FLVPlayback. Vytvoření přehrávače v prostředí Flash se dá uskutečnit dvěma způsoby. Buď zvolíte cestu s větší režii, což znamená nespolehat se na komponenty ale připravit si všechno od tlačítek, po grafiku a ovládání samostatně, nebo si vyberete klasičtější řešení a použijete předpřipravenou komponentu, kterou pak už jen budete muset doladit v jazyce ActionScript. Já jsem zvolil druhou možnost, neboť mi to přišlo vhodnější.



Obrázek 12 – Komponenta FLVPlayback

Komponenta samotná nabízí mnoho skinů²¹ a stylů. Styly jsou pro chod zásadní, protože obsahují nabízené možnosti. Můžete zvolit třeba styl, který podporuje jen přehrávání nebo zase takový, který má všechny funkce na které si jen vzpomenete. Já jsem zvolil střední cestu, kdy kromě samotného přehrávání mám ještě podporu ovládání hlasitosti, či přechod do FullScreen módu. Změna barvy, velikosti a ostatních stylistických záležitostí je samozřejmostí.

4.4.1 Syntaxe

Jak jsem již zmínil, i má video galerie je postavená na základě XML a jeho následné napojení do Flashe. Takže počáteční syntaxe je stejná, jako u výše popsáno

²¹ Měnitelný vzhled aplikace

menu. Manipulace s videem jako takovým jsem však udělal trochu odlišně. Komponentu FLVPlayback jsem si sice nahrál na pracovní plochu, ale následně jí smazal a nechal pouze v knihovně. Následně jí propojil s jazykem ActionScript a veškeré operace pak prováděl pouze s ním.

Ukázka zdrojového kódu přehrávače:

```
function makePlayer():void{
my_player = new FLVPlayback();
my_player.skin = "SkinOverPlayStopSeekFullVol.swf";
my_player.skinBackgroundColor = 0xAEBE9B;
my_player.skinBackgroundAlpha = 0.5;

my_player.autoPlay = false;
my_player.x = video_x;
my_player.y = video_y;
my_player.width = 480;
my_player.height = 270;

main_container.addChild(my_player);
my_player.source = my_videos[0].@URL;
}
```

Celá tato funkce je následně implementována do části, která se zabývá získáním dat z souboru XML, který uchovává informace o daných videích. Zde je ukázáno, jaký jsem zvolil skin a styl. Zároveň možnosti barev a rozměrů. Nejpodstatnější část se týká přiřazení přehrávače do výsledného kontejneru a zároveň načtení informací o položkách jednotlivého videa.

4.4.2 Importování

Ještě se musím zmínit o jedné věci, která je pro komponentu FLVPlayback důležitá. A tou je import potřebných tříd, nebo balíčků. To platí víceméně o většině komponent. Abych mohl vůbec přehrávač použít a ActionScript mu rozuměl, musím nejprve na začátek kódu importovat třídy nutné pro chod a správnou funkčnost. Obvykle platí, že nezkazíte nic tím, pokud toho neimportujete víc. Ale nic se nemá zbytečně přehánět.

```
import flash.text.*;
import flash.utils.Timer;
import flash.events.TimerEvent;
import fl.video.*;
import flash.display.Stage;
import flash.display.StageAlign;
import flash.display.StageScaleMode;
import flash.events.Event;
import flash.display.StageDisplayState;
```

Zde jsem ukázal svou „importovací“ část, která náleží do souboru VideoGalerie.as, který patří k video galerii s přehrávačem. Ne všechny importované věci se přímo vážou k videu, ale potřeboval jsem podporu jak zobrazení, tak i časovače a práci s textem, který používám v dané galerii.

4.5 Tweener

Navázal bych na předchozí téma o importech. Adobe Flash CS4 disponuje už v základní výbavě vším potřebným, co se od něj žádá. Má obrovskou škálu vlastních tříd. Přesto však některé nedostačují. Na internetu se za dobu fungování vytvořilo mnoho uskupení, nebo vývojářů samotných, který vlastní třídy tvoří a poskytují je ostatním. V dost případech pouze jen za nějaký peněžní obnos. Snad pro lepší funkčnost, jednoduchost jsou vytvářeny stále nové a nové. Tween je třída, kterou Flash používá už od nějaké rané verze. Slouží především k lepší práci s animacemi a hladšími přechody. Tvorba animací přes časovou osu neumožňuje tolik úprav a nedává animacím plné využití, které umožňují. Naopak Tween, který je řešen přes ActionScript umožňuje daleko lepší režii. Zajišťuje lepší a širší práci s animacemi. Podporuje časové intervaly a zlepšuje linearitu a celkově klade na animaci větší důraz. Proto ve své práci používám na podporu animací, právě Tweening.

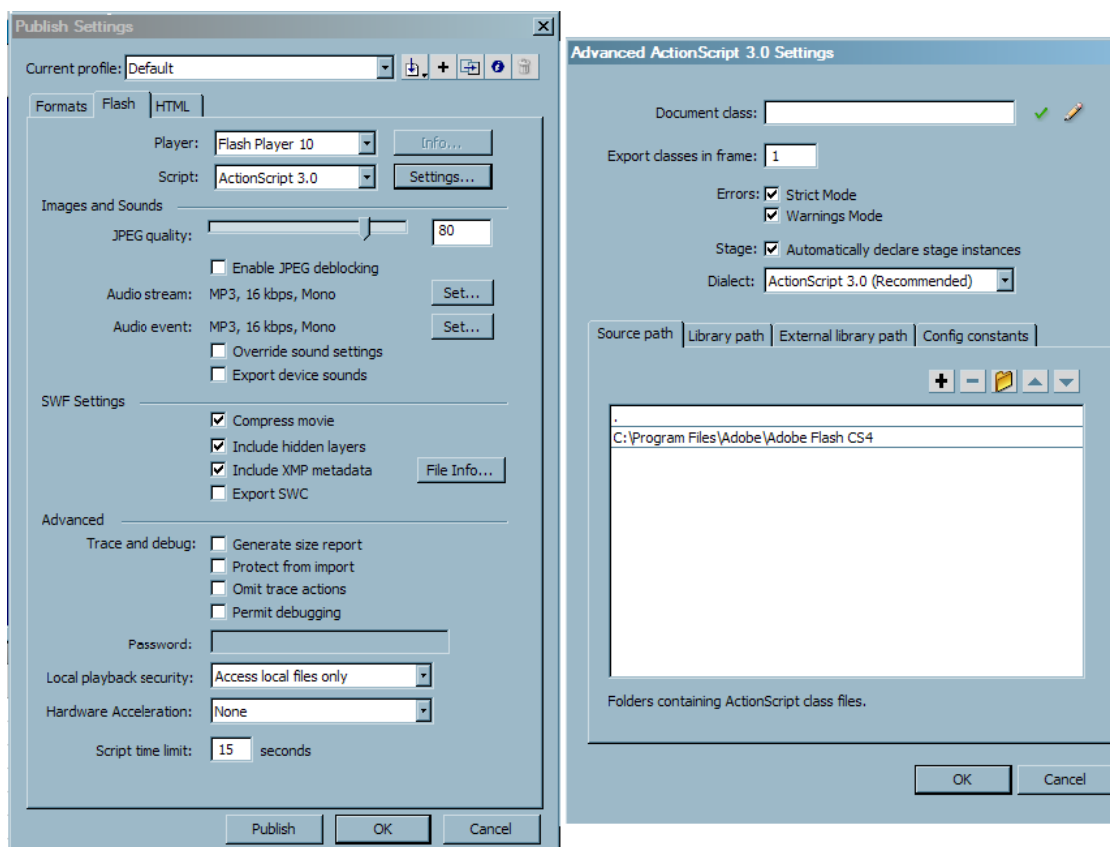
Ovšem i toto se časem rozšířilo, konkrétně ale od skupiny lidí zvenčí. Uskupení lidí nazývané GreenSock vytvořilo takzvanou Tweening Platformu, která v sobě zahrnuje projekty TweenNano, TweenLite a TweenMax. Právě poslední zmiňovaný byl využit v mé práci. A to konkrétně pro hladký přechod v rozbalovacím menu. TweenMax totiž rozšiřuje další nabídku nástrojů a úprav a už tak dělá z mocné animační a transformační třídy, třídu ještě mocnější a komplexnější.

4.5.1 Animace

Vrátím se znovu k rozbalovacímu menu, kde má třída TweenMax největší užití. Je zdepoužita především pro hladký přechod mezi položkami menu, když se menu vysouvá a nebo zasouvá. Více se vysvětlí na následující ukázce:

```
import com.greensock.*;
import com.greensock.easing.*;
import com.greensock.plugins.*;
import flash.events.*;
import com.greensock.TweenMax;
TweenPlugin.activate([GlowFilterPlugin]);
```

Zde uvádím velice potřebný import, bez kterého by nic nefungovalo. Na co se nesmí ale zapomenout je to, že TweenMax je externí třída, a jako taková se nejprve musí řádně naimportovat. Vizte zde:



Obrázek 13 – Import tříd

Na tomto obrázku je znázornění importu externích tříd. Pokud si potřebnou třídu stáhnete a někam nahrajete, Flash si jí automaticky nenajde. Proto, aby jste jí mohli používat a importovat všechny její funkce a pod-třídy, musíte k ní tímto způsobem ručně zadat cestu.

Tímto jsem si třídu připravil k použití a mohl jsem se vhrnout na psaní vlastních funkcí. Využil jsem jí při tvorbě menu, kdy se zjistí, jestli existují dané pod-položky a jestli ano, tak se menu rozvine s určitou barvou a přidáním takzvaným Glow²² efektem.

Ukázka zdrojového kódu:

```

if (selectedMenu) {
    for (var i =0; i< selectedMenu.length-1; i++) {
        TweenMax.to(selectedMenu[i],0.5,{y:87,
        glowFilter:{color:0x324df, alpha:0, blurX:0, blurY:0}});
    }
}

```

²² Přeloženo jako záře, zářit, svítit.

4.6 Odkazy

Abych zajistil hladké přepínání mezi různými kategoriemi, musel jsem zařídit nějaký chytrý systém propojení. Taková věc se řeší jako hned z prvních, když se například navrhuje menu, nebo přechod mezi tématy. Spolu s dalšími atributy u souborů XML jsem tedy přidal i atribut cesta, kde byl odkaz na jiné místo. Další práce tak zůstala na jazyce skriptovacího jazyka ActionScript, který si s tím náležitě poradil.

```
Var URLRequest:URLRequest =
new URLRequest(Smy_images[e.target.name].@FULL);
try{
    navigateToURL (URLRequest,"_blank");
}catch (e:Error){
    trace(e);
}
```

Tady je vidět řešení „přenosu“ adresy do Flashe. Konkrétně v tomto případě se jedná o klasické užité vhodné především pro web a soubory HTML.

```
var myLoader : Loader = new Loader();

var link:MenuItem = e.target as MenuItem;

var URLRequest:URLRequest = new URLRequest(link.link);

myLoader.load(urlRequest);

addChild(myLoader);
```

A tady je příklad trochu jiný od toho předešlého, kdy se toto využívá především u načítání externího obsahu a to v rámci jednoho přehrávače SWF.

Ještě je možná si všimnout, že u každého způsobu, který využívám, je důležité načtení ze souboru XML, jehož některé části se rozprostírají dále do celého kódu. V prvním případě z atributu s názvem „FULL“ a ve druhém s názvem „link“, který je však přiřazen v jiné části kódu.

5 Závěr

Cílem bakalářské práce bylo vytvořit schopný a uživatelsky přívětivý nástroj s podporou multimédií. Zároveň ho uzpůsobit tak, aby se mohl použít v praxi pro podporu výuky a kvalitního zobrazení studijních materiálů. Práce jako taková je nadále podrobena testování a teprve ostrý provoz ukáže nějaké definitivní výsledky. Práce byla velmi komplexní a rozsáhlá a jako taková nabízí mnoho úprav k vylepšení a doplnění. Pro rozšíření a její další uplatnění by bylo vhodné ještě více zpříjemnit uživatelské rozhraní, možná si i pohrát s myšlenkou větší personalizace a profilů.

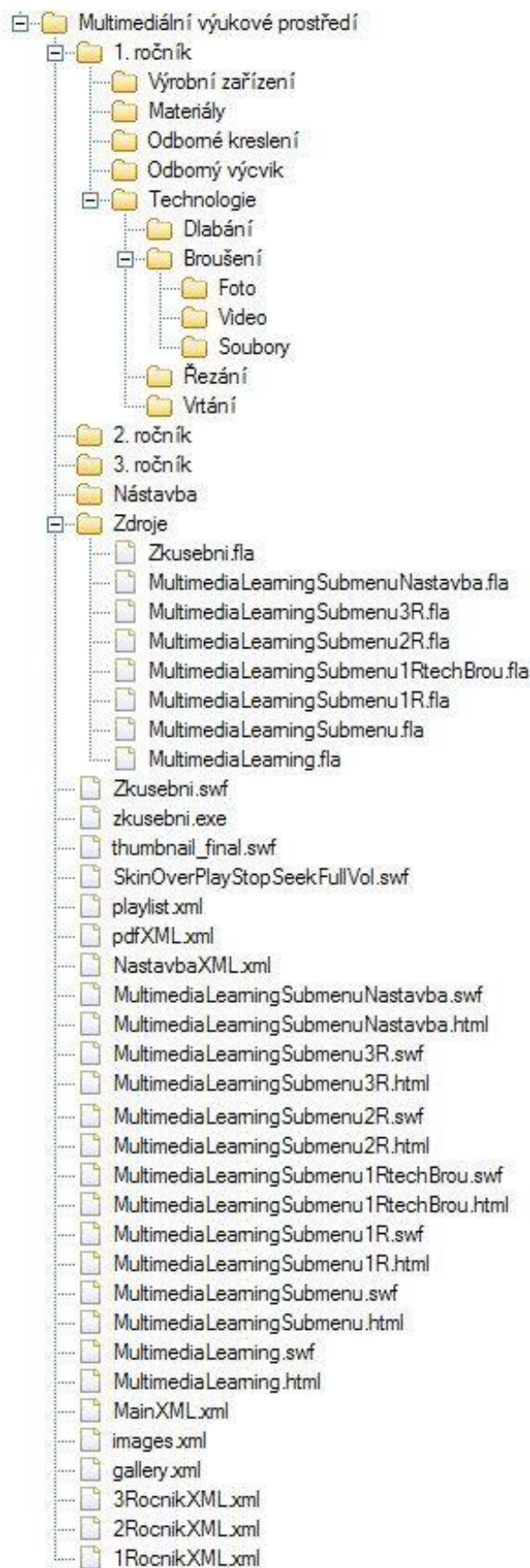
Návrh implementace a struktury odpovídal zadaným požadavkům a použité technologie jsem již dříve zmínil v práci. Technologie Flash mi umožnila vyřešit velkou spoustu problémů. Zároveň ale taky odhalila občasně neduhy, které jsou s ní spojené a které několikrát způsobili, že jsem se dostal do situací, který byly občas velmi složitě řešené.

Pokud bych měl něco napsat na úplný závěr, tak by to bylo nejspíš to, že tato práci mi dala opravdu hodně. Ještě před jejím započítím byl pro mě Flash a ostatní multimediální technologie hodně vzdálené. Díky této práci jsem se snažil proniknout do celé problematiky a řádně si jí osvojit. Naučil jsem se díky ní mnoho nového a poznal spoustu nových věcí, kterým bych se rád v budoucnu věnoval.

Zdroje

- [1] ADOBE. *Adobe Flash CS4 professional, oficiální výukový kurz*. Brno : Computer Press, 2009. 392 s. ISBN 978-80-251-2334-8.
- [2] SCHAEFFER, Mark. *Adobe Flash CS4 professional, 100 nejlepších postupů*. Brno : Computer Press, 2009. 264 s. ISBN 978-80-251-2333-1.
- [3] *Microsoft Silverlight*. Wikipedia: the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 26.2. 2007 , 6.5. 2010 [cit. 2010-05-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Microsoft_Silverlight>.
- [4] *JavaFX*. Wikipedia: the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 8. 5. 2007, 3. 5. 2010 [cit. 2010-05-09]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/JavaFX>>.
- [5] HÁJEK, Petr. *Adobe Flash vs. Microsoft Silverlight. Kdo s koho?* [online]. 22. 5. 2008 [cit. 2010-05-9]. Dostupné z WWW: <<http://www.slunecnice.cz/tipy/adobe-flash-player-vs-microsoft-silverlight/>>.
- [6] *Adobe Flash*. Wikipedia: the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 10. 1. 2002, 9. 5. 2010 [cit. 2010-05-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Adobe_Flash>.
- [7] KIRUPA. *Reading XML Files Directly*. [online]. 11. 5. 2007 [cit. 2010-05-9]. Dostupné z WWW: <http://www.kirupa.com/net/reading_xml_directly_pg1.htm>.
- [8] HRBÁČEK, Jiří. *Co umí třída Tween*. [online]. 29. 1. 2009 [cit. 2010-05-9]. Dostupné z WWW: <<http://www.flash.cz/portal/clanek.aspx?id=1327>>.
- [9] *Rich Internet application*. Wikipedia: the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 7. 11. 2004, 9. 5. 2010 [cit. 2010-05-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Rich_Internet_application>.

Příloha A – Adresářová struktura



Zdrojové soubory aplikace jsou umístěné v adresářové struktuře podle určeného umístění. Ve zkratce popíšu funkci jednotlivých složek.

- Složka **1. Ročník** – Obsahuje další podsložky, připadající danému tématu.
 - Složka **Technologie** – Například jedno z pod-témat 1. Ročníku, které obsahuje další pod-kategorie.
 - Složka **Broušení** – Například tato kategorie, v níž se nachází členění na složky foto, video a soubory.
- Složky **Dalších ročníků**– Obsahují stejnou strukturu jako ukázkový první ročník. Jen se samozřejmě liší v tématech, kategoriích a materiálech v sobě obsažených
- Složka **Zdroje** – Obsahuje všechny zdrojové soubory vytvořené v program Adobe Flash CS4
- **Kořenový adresář** – Obsahuje zdrojové a spustitelné soubory s příponami SWF, HTML a XML

Případná změna se může týkat kořenového adresáře, kdy ten může být rozdělen do více příslušných složek. Já jsem však zvolil prozatím tento způsob, neboť díky provázanosti všech zdrojových a spustitelných souborů je to podle mého efektivnější.