

Západočeská univerzita v Plzni

Fakulta pedagogická

Diplomová práce

ACTIONSCRIPT AS3

Jan Hodinář

Plzeň 2010

Prohlašuji, že jsem práci vypracoval(a) samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 2010

.....

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce Ing. Pavlu Kocurovi, CSc. za cenné rady, konzultace a čas, který věnoval mé práci.

Obsah

1.	Úvod.....	1
2.	Adobe Flash – vazba na ActionScript.....	3
2.1.	Adobe Flash obecně.....	3
2.1.1.	Platforma Flash.....	4
2.1.2.	Nevýhody Flashe	9
2.1.3.	Flash Player.....	11
2.2.	ActionScript obecně a jeho vazba na platformu Flash.....	12
2.2.1.	Základní informace o jazyce	12
2.2.2.	Vývojová prostředí.....	13
2.2.3.	Integrace jazyka do platformy.....	18
3.	ActionScript a jeho vývoj.....	19
3.1.	Historie a vývoj platformy Flash	19
3.1.1.	První kroky	19
3.1.2.	Macromedia Flash Player 2.....	20
3.1.3.	Macromedia Flash Player 3.....	21
3.1.4.	Macromedia Flash Player 4.....	22
3.1.5.	Macromedia Flash Player 5.....	23
3.1.6.	Macromedia Flash Player 6.....	24
3.1.7.	Macromedia Flash Player 7.....	25
3.1.8.	Macromedia Flash Player 8.....	26
3.1.9.	Adobe Flash Player 9.....	27
3.1.10.	Adobe Flash Player 10.....	29
3.2.	Srovnání starších verzí ActionScriptu	30
3.2.1.	ActionScript 1.0.....	30
3.2.2.	ActionScript 2.0.....	31
4.	Rozbor ActionScriptu 3.0.....	33
4.1.	Účel a prostředky	33
4.1.1.	ECMA skript.....	34
4.1.2.	Virtuální stroj.....	34
4.2.	Cíle jazyka	35
4.3.	Vlastnosti jazyka.....	36
4.3.1.	Jádro jazyka	36
4.3.2.	ECMA pro XML	38
4.3.3.	Flash Player API.....	39
5.	Využití ActionScriptu 3.0.....	42
5.1.	Pohybové efekty	42
5.1.1.	Skládací obrazy.....	42
5.1.2.	Gumový obraz.....	43
5.1.3.	Rotace objektů mimo jejich střed.....	44
5.1.4.	Carousel menu	45
5.2.	Grafické efekty.....	45
5.2.1.	Kaleidoskopické obrazy	45
5.2.2.	Efekt lupy.....	46
5.3.	Částicové efekty	47
5.3.1.	Rozklad obrazu	47
5.3.2.	Pohyb částic.....	48
6.	Závěr	49

7.	Seznam použitých zdrojů.....	50
8.	Seznam obrázků	53
9.	Seznam tabulek.....	54
10.	Seznam příloh	55
11.	Cizojazyčné resumé	56

1. Úvod

V poslední době svět výpočetní techniky stále více a více využívá softwarových výrobků firmy Adobe. Není snad na světě počítač připojený k internetu, který by nepoužíval alespoň jeden z produktů této úspěšné firmy. V době, kdy je u webových aplikací a internetu obecně kladen důraz hlavně na využití multimédií a maximální interaktivitu, vystupuje z řady aplikací firmy Adobe především multimediální platforma Flash.

Tato platforma se původně využívala k přidání animací a interaktivity do webových stránek, ovšem dnes už tento primární účel překročila a má i spoustu jiných využití, jako tvorbu internetových her, aktivní reklamy, streamování videa a nakonec tvorbu bohatých internetových aplikací¹. Klíčem k těmto rozšířeným funkcím se stal objektový programovací jazyk ActionScript, který byl vyvinut speciálně pro využití aplikací Flash. Moje diplomová práce se bude věnovat právě tomuto jazyku, který je již nedílnou součástí všech Flashových aplikací, a zvláště pak jeho nejnovější verzi.

V této práci bych chtěl nejprve vystihnout vazbu mezi Flaschem a ActionScriptem, a proto budu věnovat první část vztahu právě mezi aplikací a jazykem, který využívá. Načrtnu zde obecné možnosti Flashe a jaký vliv na tyto možnosti má právě ActionScript. První část se také okrajově zaměří na vývojové prostředí aplikace Flashe, aby si čtenář mohl lépe představit funkce, které se mu při práci s nimi zpřístupní.

Ve druhé části, když už čtenář bude znát souvislosti aplikace a jazyka, mu představím historii a vývoj všech částí celku, který zahrnuje práci s Flaschem. To se týká jak samotné aplikace Flash a firem, které ji postupně vlastnily, tak přehrávačů schopných s ní pracovat, ale hlavně jazyka samotného. Zaměřím se především na verze 1.0 a 2.0, protože tu třetí a zatím poslední rozeberu do větších detailů v další části.

Třetí část bude zajímavá hlavně z programátorského hlediska, protože se zde pokusím rozebrat ActionScript 3.0 z tohoto úhlu pohledu a prozkoumám jej jako

¹ Rich Internet Application (RIA) – je termín používaný pro webovou aplikaci, která má mnoho stejných charakteristik jako aplikace klasická. Typicky se zobrazuje pomocí speciálního prohlížeče anebo pomocí pluginu v normálním webovém prohlížeči.

objektový programovací jazyk, který se specializuje nejen na webové, ale také na klasické aplikace. Protože tato práce není učebnicí programování ani jazykovým manuálem, proberu zde hlavně změny a zlepšení oproti předchozí verzi a důvody vzniku verze 3.0. Tato část proto bude techničtějšího rázu a bude zahrnovat vše, co by si programátor začínající, ale hlavně přecházející ze starší verze, přál vědět.

V poslední části se budu věnovat především současnému využití ActionScriptu a zaměřím se proto na všechny aspekty, ve kterých se Flash s podporou tohoto jazyka využívá a ve kterých vyniká. Tuto kapitolu doplním konkrétními praktickými ukázkami toho, čeho všeho nejnovější verze ActionScriptu dokáže jednoduše a efektivně dosáhnout a k čemu se tedy dnes často využívá.

2. Adobe Flash – vazba na ActionScript

Jak už jsem napsal v úvodu, v této části čtenářům, s tematikou neseznámeným, objasním co je to Adobe Flash a ActionScript a dále jakým způsobem spolu tyto dvě složky spolupracují k dosažení kýženého výsledku. Do popisu Flashe zde také zahrnu jeho vývojová prostředí a s čím vším může uživatel při práci s nimi počítat.

2.1. Adobe Flash obecně

Pojmem Flash může být myšleno více věcí. Prvním a nejpřesnějším významem je již v úvodu zmiňovaná multimediální platforma, avšak tím to zdaleka nekončí. Když se vás někdo zeptá, zda máte na počítači nainstalovaný Flash, máte několik možností, jak odpovědět a těžko budete tazateli vysvětlovat, že máte nainstalovanou multimediální platformu.

Tazatel nejspíše touto otázkou myslí, zda váš počítač je schopný přehrávat Flashové soubory anebo zda váš prohlížeč je schopen zobrazovat webový obsah vytvořený ve Flashi. Druhým významem slova Flash tedy může být také takzvaný Flash Player. Což je software ve formě pluginu² právě pro zobrazování Flashových filmů či animací přímo na PC³ a nebo například na zařízení typu mobilního telefonu či handheldu⁴, o noteboocích ani nemluvě.

Třetím a posledním významem slova Flash může být aplikace, která vytváří již zmiňované soubory, se kterými pak platforma pracuje. Celý název stejně jako název platformy je Adobe Flash, i když dříve celou platformu vlastnila firma Macromedia, a proto je možné se ještě na některých starších webových stránkách setkat s pojmem

² Je malý počítačový program, který po doinstalaci značně rozšiřuje schopnosti většího programu.

³ PC – Personal Computer – osobní počítač

⁴ Je lehký, malý a skladný přenosný počítač s vlastním napájením.

Macromedia Flash. Tato aplikace je zároveň vektorový⁵ grafický program a vývojové prostředí aplikací, pracující na platformě Flash a obsahující jazyk ActionScript.

2.1.1. Platforma Flash

Několikrát jsem již v této práci použil slovo platforma, které je třeba vysvětlit, aby bylo jasné v kontextu s Flashem. Platformou je v počítačové terminologii míněno pracovní prostředí jak po stránce hardware, tak i software (většinou framework⁶). Toto prostředí umožňuje bezproblémovou činnost programů. Mezi typické platformy patří zejména hardwarová architektura počítače, operační systém nebo programovací jazyky s jejich knihovnami. Flash je tedy platforma sloužící k výrobě a následné podpoře chodu multimediálních webových i newbových aplikací.

Obecné možnosti

První účel Flashe byla implementace vektorové grafiky do webových stránek, ovšem dnes už toho umí mnohem více. Toto téma rozeberu více v dalších kapitolách. Obecně shrnuto Flash manipuluje s vektorovou i rastrovou grafikou, aby dosáhl animace textu, kreseb i statických obrázků. Podporuje oboustranné streamování⁷ zvuku i videa. Může zaznamenat uživatelský vstup jak pohyby myši, tak pomocí klávesnice či mikrofonu nebo kamery. Poslední dobou se hlavně používá k tvorbě malých i větších webových aplikací, které slouží jako interface webových stránek, webové hry anebo reklama. Nebudu zde jmenovat všechny servery, které technologii Flash využívají. Jedním příkladem za všechny je například server „youtube“, který by bez Flashové technologie streamování videa nebyl schopen dopravovat svým uživatelům doslova miliony audiovizuálních záznamů jen pomocí prohlížeče.

⁵ Vektorové grafické programy pracují hlavně s vektoru grafikou tj. s grafikou složenou z geometrických obrazců nikoliv z rastrové bitmapy, jak je tomu u rastrové grafiky.

⁶ je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů a aplikací. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.

⁷ je technologie kontinuálního přenosu audiovizuálního materiálu mezi zdrojem a koncovým uživatelem

Formáty souborů

Flash má několik různých formátů souborů, které s ním souvisejí, ovšem nejdůležitější a nejpoužívanější z nich jsou SWF a FLV. Flashové soubory s příponou .swf (Shock Wave Flash) mohou být také nazývány „Flashovými filmy“ nebo „Flashovými hrami“. Tyto soubory mohou být vloženy přímo do webových stránek nebo být přehrány samostatně pomocí nainstalovaného Flash Playeru. Existují také varianty s příponou .exe, které jsou samospustitelné pod OS⁸ Windows, protože mají přehrávač v sobě již integrovaný.

Tento formát je částečně otevřený. To znamená, že některé prvky jeho struktury jsou známy a je dovoleno je generovat jinými prostředky než těmi od firmy Adobe. Formát pracuje se spojením multimedií a vektorové grafiky, které bylo od začátku zamýšleno tak, aby bylo co objemově nejmenší a tudíž se hodilo k publikaci na internetu. Tento úmysl se podařilo dotáhnout velmi daleko a dnes je SWF naprosto dominantním formátem používaným k zobrazení „animované“ vektorové grafiky. Stejně tak se velmi často používá k tvorbě aplikací a webových her nebo reklam využívajících při tom výhod programovacího jazyka ActionScript. Podle nezávislé studie firmy Millward Brown má dnes 99% uživatelů webu ve svých prohlížečích nainstalován plugin umožňující SWF soubory přehrávat.

Soubory typu FLV jsou stejně jako soubory SWF kontejnerové⁹. Byly vytvořeny za účelem snadno a rychle dopravit uživateli video přes internet. Budu se jimi podrobněji věnovat dále, až se budu zabývat Flash videem a způsobem, jakým pracuje. Nyní bych rád zběžně zmínil ostatní formáty souborů, které sice nejsou tak rozšířené a běžný uživatel se s nimi velmi často neseťká, ovšem pro vývoj a práci celé platformy mají nezanedbatelný význam.

⁸ Operační systém

⁹ Je formát souborů umožňující do sebe pojmout soubory jiného typu a účinně je zkombinovat. Zpravidla se jedná o multimediální soubory, ale není to pravidlem.

Ostatní formáty souborů související s Flashem

V tabulce 1 jsem zmínil a krátce popsal většinu souborů, se kterými celá platforma Flash pracuje.

Přípona	Vysvětlení
.swf	Kromě toho uvedeno výše jsou swf soubory kompletní, zkompileované a většinou publikované soubory, které nemohou být již změněny vývojovým nástrojem Flash. Ačkoliv za tímto účelem vzniklo mnoho .swf dekompilátorů ¹⁰ . Snaha „otevřít“ swf soubor ke zpětné editaci pomocí vývojového nástroje Flash je však částečně úspěšná a některé vlastnosti a nastavení mohou být ze souboru získány.
.FXG	FXG je spojený xml formát, který je vyvíjen firmou Adobe pro Flex, Flash, Photoshop a také jiné aplikace.
.fla	.fla soubory jsou zdrojové soubory pro Flashové aplikace. Vývojový nástroj Flash je může editovat a také z nich kompilovat .swf soubory. V současnosti je zdrojový soubor Flash binární soubor založený na formátu Microsoft Compound File ¹¹ . Ve Flashi verze CS5, která zanedlouho vyjde, jsou však fla formáty založeny na principu archivu zip se strukturou na principu XML, podobně jako je tomu u Office 2007.
.xfl	.xfl soubory jsou založeny na struktuře XML a jsou ekvivalentem binárních souborů .fla. Vývojový nástroj Flash používá xfl soubory jako výměnný formát ve Flashi CS4. Tyto soubory jsou určeny pro import do Flashe a jsou exportovány Adobe aplikacemi InDesign a AfterEffects. Ve Flashi verze CS5 jsou xfl soubory klíčovými soubory, které otvírají nekomprimované soubory fla, ve kterých je obsažena hierarchie XML a binární soubory.
.as	.as soubory obsahují zdrojový kód jazyka ActionScript v jednoduchých zdrojových souborech. Fla soubory mohou také obsahovat tento zdrojový kód, ovšem soubory .as jsou často separované z důvodu hierarchie kódu anebo kvůli možnosti rozdělit určité části kódu do několika verzí. Tyto soubory někdy také používají příponu .actionscript.
.mxm1	.mxm1 soubory jsou použity ve spojení se soubory ActionScriptu (a .css soubory) a nabízejí značkovací syntaxi podobnou té jako má jazyk HTML. Používají se k návrhování GUI ¹² ve Flexu ¹³ .
.swd	.swd soubory jsou dočasné soubory potřebné k ladění a debugu ¹⁴ Flashových aplikací. Jakmile je vývoj aplikace dokončen, tyto soubory již nejsou potřebné a mohou se smazat.

¹⁰ Dekompilace je opačný proces kompilace. Jedná se o snahu vytvořit programový kód z již vytvořeného výsledného souboru (například .exe). Je to forma zpětného inženýrství.

¹¹ Je datový formát určený k ukládání bezpočtu souborů a streamů do jednoho souboru na disku.

¹² Graphical User Interface – jedná se v podstatě o to, co uživatel v aplikaci vidí a s čím může nějak operovat.

¹³ Flex je vývojový software firmy Adobe určený k tvorbě bohatých internetových aplikací.

¹⁴ Odstranění chyb.

.asc	.asc soubory obsahují ActionScript na straně serveru, což je použito k vývoji efektivních a flexibilních klient-server Flash aplikací
.abc	.abc soubory obsahují zkompileovaný kód ActionScriptu a jsou používány virtuálním strojem ActionScriptu ¹⁵ .
.flv	.flv soubory jsou Flash video soubory stvořené aplikací Adobe Flash, ale i jinými. Audio a video data v FLV souborech jsou zakódována stejným způsobem jako je tomu u souborů SWF.
.f4v	.f4v soubory jsou obdobou klasických MP4 souborů mohou být přehrávány Flash playerem 9 update 3 a výše. F4V je druhý kontejnerový formát na přehrávání Flash videa a značně se liší od FLV formátu, protože je založen na ISO formátu pro obecná média.
.f4p	.f4p soubory jsou F4V soubory obsahující technologii Digital Rights Management ¹⁶ .
.f4a	.f4a soubory jsou F4V soubory obsahující pouze audio streamy.
.f4b	.f4b soubory jsou určeny k vydávání audio knih.
.swc	.swc soubory jsou používány pro distribuci komponent. Obsahují zkompileovaný klip, programový kód komponenty napsaný v ActionScriptu a další soubory, které popisují danou komponentu.
.jsfl	.jsfl soubory jsou používány k tomu, aby dodaly různé funkce do vývojového prostředí Flashe. Obsahují kód JavaScriptu ¹⁷ a přístup k Flash JavaScript API ¹⁸ .
.swt	.swt soubory jsou jistou formou šablony pro SWF soubory.
.flp	.flp soubory jsou v podstatě soubory XML určené k tomu, aby spojily dohromady všechny soubory a zdroje používané v projektech vývojové aplikace Flash. Flash projekty umožňují uživateli seskupit libovolný počet jakéhokoliv typu souborů k lepší organizaci, tvorbě a kompilaci výsledných Flash souborů (např. SWF).
.spl	Jedná se o dokumenty aplikace FutureSplash ¹⁹ .
.aso	.aso jsou cache soubory používané při vývoji Flash aplikací a obsahují zkompileovaný kód ActionScriptu. ASO soubor je znovu vytvořen a přepsán v okamžiku, kdy je detekována změna ve třídě, která se souborem koresponduje. Občas se stávalo, že vývojové prostředí Flashe nerozeznalo změnu a soubory musely být smazány manuálně.
.sol	.sol soubory jsou vytvářené Flash Playerem aby v sobě držely Local Shared Objects. Což je v podstatě Flash obdoba klasických Cookies.

Tabulka 1 Typy souborů platformy Flash

¹⁵ ActionScript Virtual Machina (AVM) – je virtuální stroj starající se o spouštění programů napsaných v ActionScriptu. Zatím má 2 verze.

¹⁶ Jsou technologie zabezpečení přístupu k souborům, které mohou být používány výrobci hardwaru, vydavateli a vlastníky práv. Díky nim lze docílit limitovaného používání digitálního obsahu a různých zařízení.

¹⁷ Objektově orientovaný skriptovací jazyk používaný především ve web aplikacích.

¹⁸ Application Programming Interface – je rozhraní poskytující knihovny funkcí různým aplikacím.

¹⁹ Jedná se o předchůdce Flashe. Aplikace byla představena v roce 1995.

Flash Audio a Video

Před zavedením standardu HTML5 záviselo zobrazení videa na webových stránkách na platformě prohlížeče a vznikaly problémy s tím spojené. Problém byl v tom, že neexistoval standard pro zobrazení videa přes web a také v tom, že nebyl stanoven žádný běžně používaný video kodek²⁰. Díky tomu se vývojáři webových stránek museli obrátit na technologii Flashe, která ovšem byla licencovaná. Tímto vznikl mezi uživateli internetu nechvalně proslulý efekt toho, že multimédia vložená do stránek pomocí Flashe, nešla spustit, aniž by uživatel měl nainstalován prohlížeč obsahující Flash plugin anebo přímo externí Flash Player. Nicméně to také způsobilo, že nástup Flashe v pravou chvíli zapříčinil jeho současnou dominanci na trhu.

Jak už bylo naznačeno v tabulce 1, soubory typu Flash video jsou kontejnerové formáty určené k rychlému a snadnému doručení videa přes internet a jdou přehrávat pouze pomocí Flash Playeru nebo jeho potřebného pluginu v prohlížeči. Soubory mají přípony .flv a .f4v a samy mohou být obsaženy v souboru typu SWF. Rozdíl mezi těmito formáty je dobře patrný z předchozí tabulky.

I když je kontejnerový formát pro Flash Video otevřený, většina kompresních formátů v něm obsažených jsou patentovány, a tudíž veřejnosti nedostupné. Flash Video .flv soubory většinou mají obsah komprimován některým z potomků kodeku Sorenson Spark²¹ nebo VP6²². V poslední době se také často objevuje obsah zakódovaný ve standardu H.264²³.

²⁰ Jedná se o zařízení nebo software umožňující kompresi a dekompresi videa.

²¹ Sorenson Spark je licencovaný digitální video kodek užívaný přehrávačem QuickTime firmy Apple a Flaschem firmy Adobe.

²² VP6 je licencovaný video kodek vyvinutý firmou On2 Technologies. Je to potomek dřívějších kodeků VP3 a VP5 a je běžně používán aplikacemi firmy Adobe.

²³ H.264/AVC/MPEG-4 část 10 je jeden z nejnovějších standardů pro komprimování videa.

Co se týče zvuku, je většinou komprimován do formátů MP3 a nebo AAC²⁴, i když podporuje i kodeky typu ADPCM²⁵, Nellymoser²⁶ a Speex²⁷. Jelikož většina Flash souborů jsou kontejnerového formátu, je až na výjimky audio stopa v souboru uložena samostatně a dá se tak bez problémů oddělit.

2.1.2. Nevýhody Flashe

Samozřejmě i Flash má několik neduhů. Většina z nich vyplývá z toho, že je celá platforma licencovaná, a tudíž nejsou k dispozici zdrojové kódy, ovšem dost kritiky také samozřejmě směřuje od konkurenčních firem a nezávislých expertů.

Špatné zabezpečení

Flash za dobu své existence posbíral pěknou řádku chyb v zabezpečení a dospělo to až do stádia, kdy US-CERT, což je zvláštní tým amerického ministerstva pro národní bezpečnost zabývající se kybernetickou bezpečností, doporučil Flash vůbec nepoužívat anebo ho v prohlížečích zablokovat využívajíc NoScript, což je program k tomu určený. Toto doporučení podpořilo i několik nezávislých expertů na bezpečnost na internetu.

K 27. dubnu 2010 měl Flash od doby svého vzniku na kontě 75 záznamů v systému CVE²⁸. Z čehož 34 bylo vyhodnoceno jako vážných a 40 jako střední závažnosti. Na začátku roku 2010 se Adobe oficiálně omluvilo za to, že nechalo a neopravilo známé kritické chyby ve Flashi, které byly v systému až jeden rok.

²⁴ Advanced Audio Coding (AAC) – je standardizovaný formát ztrátové komprese digitálního zvuku.

²⁵ Adaptive DCPM (differential pulse-code modulation) – kompresní metoda, která je založena na vyspělé formě převodu analogového signálu na digitální. Používá se například při internetovém telefonování.

²⁶ Nellymoser Asao Codec – je licencovaný jednokanálový mono audio kodek vyvinutý pro nízké přenosové rychlosti.

²⁷ Je otevřený formát audio komprese specializovaný na kompresi lidského hlasu.

²⁸ Common Vulnerabilities and Exposures – je systém k porovnávání a rozpoznávání chyb a náchylností v zabezpečení softwaru.

Výkon a nestabilita

Jakýkoliv přehrávač Flashe, který je vložený do internetového prohlížeče, musí být schopen renderovat²⁹ video na vrchu jiných aplikací, či grafických komponent anebo dokonce na jiném videu. Z čehož plyne, že i hardwarem posílené renderování videa není tak přímočaré jako u klasického přehrávače multimédií. Proto, i když přehrávače Flashe pouze zobrazují video, mají mnohem větší nároky na systémové zdroje než klasický přehrávač zaměřený pouze na přehrávání videa.

Srovnávací testy ukázaly, že výkon Flashe je mnohem vyšší na počítačích s OS Windows oproti Mac OSX³⁰ a nebo Linuxu. Steve Jobs, výkonný ředitel společnosti Apple, prohlásil, že pokud počítač od firmy Apple „spadne“, může za to ve více než polovině případů právě Flash. Navíc pak v dubnu 2010 udal 5 důvodů, proč mobilní zařízení firmy Apple nebudou Flash podporovat a raději se zaměří na HTML5.

1. HTML5 je otevřené, Flash je technologie vlastněná a plně kontrolovaná společností Adobe.
2. Flash není podle Jobse bezpečný.
3. Flash má na mobilních zařízeních často problémy se stabilitou a výkonem.
4. Flash na mobilních zařízeních spotřebovává více baterie, mohou za to starší kodeky, ve kterých se nachází většina flashového videa.
5. Flash byl navržen pro ovládání počítačovou myší. iPhone, který nezná koncept kurzoru a jeho polohy, tudíž s touto technologií není kompatibilní.

²⁹ V tomto kontextu je video renderingem zamýšlena dekomprese, složení scény a plynulé zobrazení videa. Na rozdíl od 3D grafiky, kde se jedná o proces tvorby obrazu z počítačového modelu.

³⁰ Operační systém firmy Apple

Invaze do soukromí

Stejně jako HTTP cookie³¹, Flash cookie (známa jako Local Shared Object) může být použita k tomu, aby uložila uživatelská data. Nezávislou studií bylo zjištěno, že více než 50 procent stránek využívajících Flash, také využívá možnosti skladovat Flash cookies na uživatelských počítačích do své vlastní cache. Tyto soubory mohou být použity k tomu, aby pomocí nich byla vysledována uživatelova aktivita na internetu. Přes tento fakt drtivá většina stránek na toto uživatele neupozorňuje a neumožňuje jim tuto skutečnost nijak ovlivnit. Většina dnešních prohlížečů má funkce, které mohou zakázat používání anebo úplně zablokovat cookies. Tyto funkce a také možnost vymazání klasických cookies se však na Local Shared Objects nevztahuje, a proto uživatel, ač přesvědčen že smazal všechna vystopovatelná data ze své počítače, mnohdy zanechá alespoň historii stránek využívajících Flash. Local shared Objects mohou být vymazány v nastavení manažeru webových aplikací v nastavení Flash Playeru anebo pomocí jiných nástrojů k tomu určených.

2.1.3. Flash Player

Adobe Flash Player, jak už bylo naznačeno výše, je software pro sledování animací, filmů a spouštění aplikací vytvořených na platformě Flash. Tento software se distribuuje ve formě pluginů, které mohou být použity v jiných aplikacích, nejčastěji webových prohlížečích. Flash Player přehrává primárně soubory typu SWF, které jsou vytvořeny vývojovými prostředími Adobe Flash, které je součástí balíku Adobe Creative Suite, Adobe Flex, což je sada nástrojů pro tvorbu bohatých internetových aplikací anebo jinými nástroji firmy Adobe, které mají podporu exportu do SWF. Poslední možností jsou pak nástroje jiných firem, kterými se ovšem v této práci nebudu zabývat, protože z hlediska ActionScriptu jsou irelevantní.

Flash Player byl původně navržen k zobrazování dvourozměrných animací, nicméně se za dobu svého vývoje stal schopný podpory bohatých internetových aplikací a streamování audia a videa. Používá primárně vektorovou grafiku, čímž minimalizuje velikost souborů a snižuje čas načítání a tok dat po internetu. Konkrétními příklady

³¹ Jedná se o typ souboru, ve kterém si webové stránky na uživatelském počítači uchovávají potřebná data. Jde například o nastavení stránky, nákupní košík či identifikaci počítače.

využití v dnešní době jsou internetové hry, reklamy, animace, video na internetu a grafické rozhraní webových stránek. Podrobněji se jeho praktickým použitím spolu s jazykem ActionScript 3.0 budu zabývat v poslední části této práce.

Flash Player je dostupný jako plugin pro nové verze webových prohlížečů Internet Explorer, Mozilla Firefox, Opera a Google Chrome. Adobe tvrdí, že každá verze pluginu je zpětně kompatibilní s tou předešlou až na pár výjimek, které se týkaly zabezpečení ve verzi 10. Tato zatím poslední verze 10 je určená pro operační systémy Windows 2000 a novější a pro Linux, Solaris a Mac OS X. Flash Player má také odlehčenou verzi Lite, která je určená pro miniaturní počítače a kapesní zařízení. Tato verze však většinou nemá všechny schopnosti verzí plných, a proto funguje s některým obsahem jen v omezené míře. Při vývoji aplikací pro tato zařízení tedy na toto musí být brán zřetel.

2.2. ActionScript obecně a jeho vazba na platformu Flash

Platforma Flash je dnes již velice rozsáhlá a zahrnuje několik oblastí vývoje aplikací a jejich použití. Nejhlavnější důraz je stále kladen na internet a využití na webových stránkách, avšak toto byl primární účel vývoje platformy, dnes to už zdaleka není vše.

Jak se Flash postupně vyvíjel a hledala se pro něj nová a nová uplatnění, začaly se jednotlivé oblasti využití platformy rozcházet. Vznikala nová vývojová prostředí určená speciálně pro určité oblasti a platforma už zdaleka nebyla tak uniformní jako tomu bývalo. I přes toto rozdělení stále zůstaly dva prostředky, které ji pojí dohromady. První z nich je již zmíněný Flash Player, který je stále hlavní prostředek k zobrazování veškeré tvorby, která je na platformě postavena. Druhým je právě jazyk ActionScript, který je součástí přehrávače a ve kterém se dá programovat veškerý obsah platformy.

2.2.1. Základní informace o jazyce

ActionScript je původně skriptovací, a dnes již plně programovací, jazyk vyvinutý firmou Macromedia a následně převzatý firmou Adobe. Jedná se o dialekt³² ECMA³³

³² Znamená, že má stejnou syntaxi a sémantiku jazyka.

skriptu, který byl základem pro několik programovacích jazyků a o kterém budu mluvit dále ve třetí části. Byl navržen Gary Grossmanem a jeho verze 1.0 se objevila v roce 2000. Myslím, že není důležité rozepisovat se o jeho účelu, protože se naprosto shoduje s účelem celé platformy a vylepšení schopností přehrávače bylo docíleno kromě přidání nových funkcí především také vývojem a vylepšením jazyka v dané verzi.

K dnešnímu datu existují tři plné verze tohoto jazyka. ActionScript urazil dlouhou cestu od jednoduchého skriptovacího jazyka určeného k vylepšení interaktivnosti animací k dnes již složitému programovacímu jazyku, který se používá ve všemožných oblastech.

Jazyk je multiplatformový³⁴ a používá se ve všech vývojových prostředích a softwarových balíčcích platformy Flash. Je zakomponován do všech SWF souborů a s novou verzí jazyka je nutné vylepšit i Flash Player, protože jeho verze mezi sebou nejsou kompatibilní a nelze zaměnit kód jedné verze za kód druhé.

2.2.2. Vývojová prostředí

Jak se zaměření platformy Flash přesouvalo čistě od animací a multimédií také k bohatým internetovým aplikacím, jediný vývojový nástroj, který byl v té době zaměřený právě především na vektorovou grafiku a video, přestal stačit. Platforma jako taková byla představena v roce 2004 a tehdejší verze vývojového nástroje už všem jejím požadavkům úplně nevyhovovala. Proto vznikly další nástroje a softwarové balíčky, které zde alespoň zběžně popíši.

Adobe Flash CS3 Professional

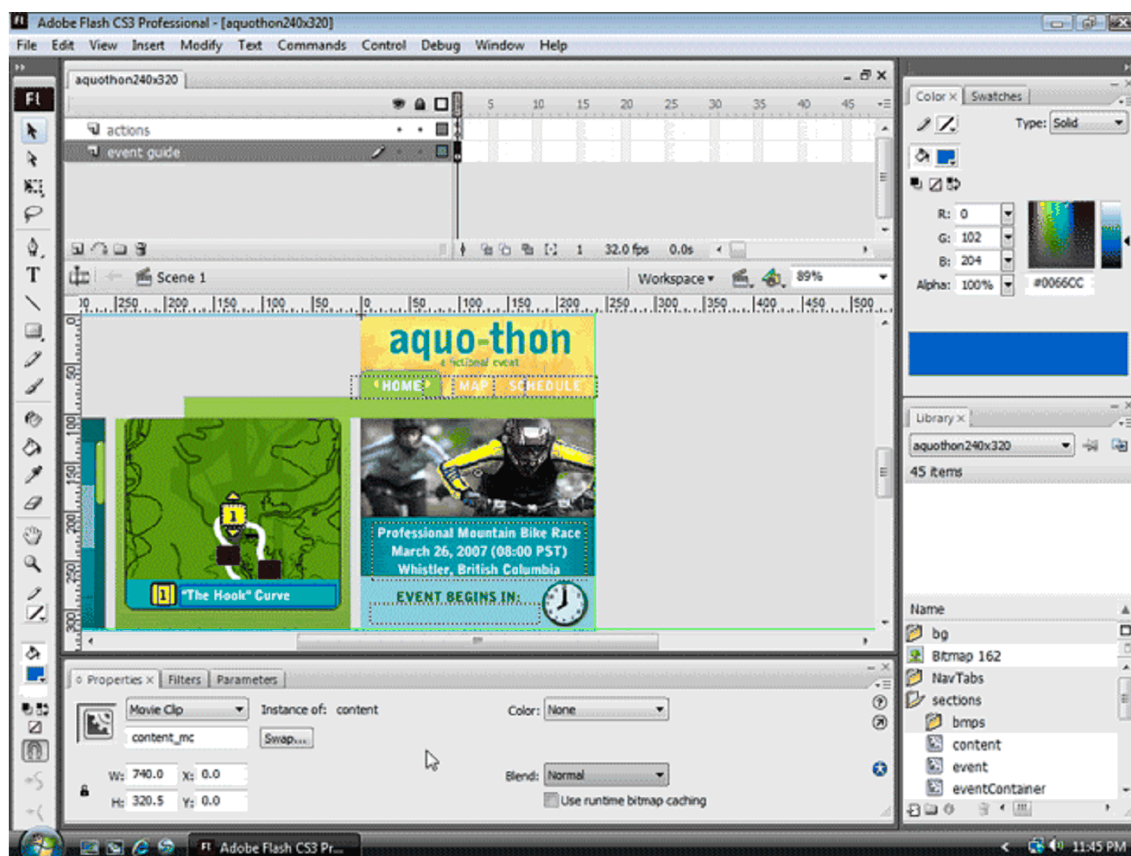
Je následovník celé řady vývojových nástrojů a autorských prostředí Flashe. Tato aplikace je potomek toho, co prvotně vzniklo jako vývojový nástroj pro Flash ještě předtím, než vznikla samotná platforma Flash. Tento nástroj následuje v duchu tradice a zaměřuje se především na animace, audio a video a menší aplikace používané na webu a

³³ Je skriptovací jazyk vyvinutý firmou Ecma International

³⁴ Jazyk je multiplatformový, pokud je použitelný na více operačních systémech, přičemž se očekává naprosto stejná funkčnost.

slouží také jako vektorový grafický editor. Novější verze tohoto nástroje se pak používají také k výrobě aplikací pro mobilní zařízení. Je součástí balíčku Adobe Creative Suite a většinou s novou verzí balíčku vyjde i nová verze tohoto nástroje.

Popisovat tady přesně jeho interface nemá cenu, protože je plně nastavitelný, a každý uživatel si ho proto může upravit k obrazu svému. Nicméně popíši hlavní části, ze kterých se skládá. První je většinou ve středu obrazovky a jedná se o plochu animace či programu. Můžeme na ni umisťovat komponenty a vidíme na ní celkovou kompozici obrazovky. Další, v tomto programu velice důležitou částí, je časová osa, na které můžeme posunovat jednotlivé snímky a vidíme pomocí ní průběh animace a jak se vyvíjí. Většinou nalevo pak bývá panel nástrojů s funkcemi grafického editoru. Napravo se nalézá seznam objektů na ploše a lze zde upravovat jejich základní parametry. A naposledy dole se nachází parametry aplikace jako celku a tuto část lze přepnout tak, aby zobrazovala kód aplikace psaný v ActionScriptu. Na obrázku 1 je zobrazen náhled právě z této aplikace.



Obrázek 1 Vývojové prostředí Adobe Flash CS3

Adobe Flex

Adobe Flex je softwarový vývojářský balíček sloužící k vývoji a podpoře běhu bohatých internetových aplikací založených na platformě Flash. Flexové aplikace mohou být napsané pomocí vývojového nástroje Adobe Flash Builder anebo pomocí volně šiřitelného kompilátoru poskytnutého firmou Adobe zdarma. Flex tedy jako takový není sám o sobě vývojářský nástroj, je to spíše soustava nástrojů a komponent, nicméně je jako celek určený k vývoji Flash aplikací a proto se do této kategorie hodí.

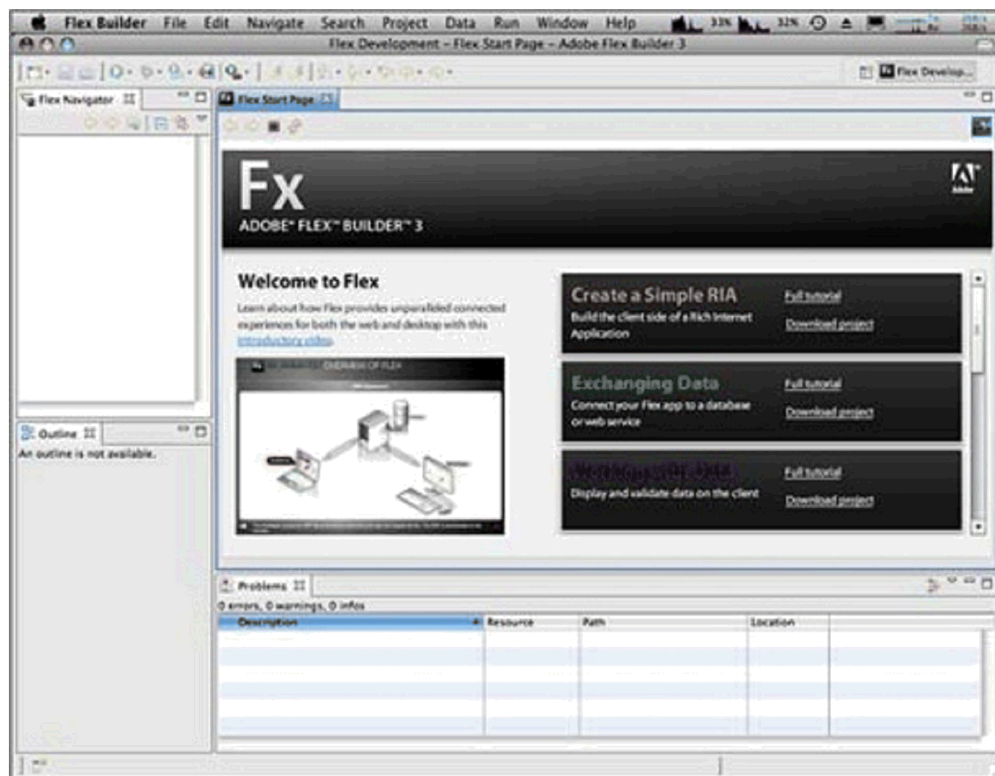
Programátoři klasických aplikací původně měli problém s modelem aplikace postavené na animaci tak, jak byla platforma Flash originálně navržena. Flex se snaží tento problém minimalizovat tím, že poskytuje programový model podobný tomu, na který jsou tito programátoři zvyklí. MXML jazyk, což je značkovací jazyk postavený na jazyce XML, nabízí jednoduchý způsob, jak postavit grafické rozhraní aplikace. Interaktivita je dosažena pomocí zabudovaného jazyka ActionScript a Flex balíček přichází s celou řadou komponent, které pomůžou při tvorbě rozhraní, jako jsou tlačítka, formuláře, grafy, datové stromy a jiné komponenty webových služeb.

Nejvíce se k tvorbě aplikací pomocí Flexu používá vývojové prostředí Flash Builder, což je integrované vývojové prostředí postavené na platformě Eclipse³⁵. Flash Builder obsahuje kromě klasického editoru MXML a ActionScript kódu také takzvaný WYSIWYG editor. Zkratka znamená What You See Is What You Get neboli co vidíš, to dostaneš. Jedná se o druh editoru, ve kterém se přímo skládají grafické komponenty a podle toho, jak je uživatel složí, editor vygeneruje kód, který toto složení popíše. Flash Builder také obsahuje zabudovaný debugger³⁶, pomocí něhož mohou programátoři sledovat průběh a hodnoty proměnných programu krok po kroku. Flash Builder se dříve jmenoval Flex Builder, avšak došlo ke změně jména, aby bylo jasně vidět spojení

³⁵ Jedná se multijazykové vývojové prostředí, které je postavené univerzálním způsobem tak, aby obsloužilo co největší počet jazyků. Toho je docíleno značně propracovaným systémem pluginů.

³⁶ Debugger je obecně počítačový program, který se používá k testování a odhalení chyb v jiných programech.

s platformou Flash a také rozlišení tohoto vývojového nástroje od celého balíčku Flex. Ukázku z nástroje verze 3 tehdy ještě pojmenovanou Flex Builder můžete vidět na obrázku 2.



Obrázek 2 Flex Builder verze 3

Adobe Integrated Runtime

Adobe Integrated Runtime také známé jako Adobe AIR je multiplatformové rozhraní vyvinuté firmou Adobe za účelem výroby a běhové podpory pro bohaté uživatelské aplikace, které mohou být použity jako aplikace klasické. Nejedná se tedy opět tak o vývojové prostředí v klasickém slova smyslu, jako spíše tentokrát o rozhraní podporující spuštění a běh bohatých internetových aplikací vyvinutých pomocí Flashe, Flexu, ActionScriptu, HTML a JavaScriptu.

Tyto aplikace jsou sice prvotně určeny pro internet, ovšem pokud jsou stvořeny pro Integrated Runtime, chovají se víceméně jako klasické programy. Adobe tento produkt profiluje jako prostředí pro bohaté internetové aplikace ovšem bez podpory webového prohlížeče. Toto paradigma způsobuje výhody i nevýhody. Aplikace určená pro

prohlížeč nepotřebuje instalaci, na druhou stranu aplikace určená pro AIR musí být dodána jako instalační balík, musí být digitálně podepsána a musí se nainstalovat na lokální disk. Což na druhou stranu poskytuje aplikaci přístup k datům v uživatelské počítači, který klasické internetové aplikace mají značně omezený, čímž dovoluje aplikaci větší flexibilitu.

AIR se používá hlavně k tvorbě menších pomocných aplikací, které uživatel může mít zobrazeny na ploše anebo her. V lednu 2009 Adobe vydalo prohlášení, že si AIR nainstalovalo již 100 miliónů uživatelů a že většina instalací byla zapříčiněna nutností ke spuštění AIR aplikace instalované uživatelem. Pravdou však je, že tento počet pravděpodobně zapříčinilo i to, že bylo rozhraní AIR v té době součástí instalace Adobe Acrobat Reader 9, což je oblíbená aplikace pro zobrazování PDF souborů. Adobe přidává AIR do této instalace dodnes bez možnosti tuto část instalace odmítnout. Na obrázku 3 je vidět souhrn několika aplikací určených pro AIR.



Obrázek 3 Ukázka aplikací Adobe Integrated Runtime

2.2.3. Integrace jazyka do platformy

ActionScript jako takový vznikl pro použití ve vývojovém prostředí Adobe Flash, který s verzí CS3 odpovídá nejnovější verzi ActionScriptu 3.0. Ovšem, jak už jsem naznačil výše, se vznikem platformy Flash vznikly i jiné nástroje, které si ActionScript začaly „půjčovat“. Mezi ty nejvýznamnější patří popsané v předešlé kapitole. S tímto ovšem nastala otázka, zda je jazyk mezi těmito nástroji přenositelný anebo zda je třeba brát ohled na určité problémy s tímto spojené.

ActionScript je jazyk společný pro celou platformu Flash, to znamená, že zkušenosti a návyky s jazykem budou aplikovatelné kromě například CS3 i v jiných oblastech platformy a rozvíjejí tak dosah a možnosti programátora. Ovšem existuje tu pár rozdílů, kterým je třeba napřed porozumět, než je možné se pustit do programování mezi více aplikacemi platformy. Zde je několik příkladů těchto rozdílů.

V první řadě Flash a Flex mají rozdílné kompilátory, takže neexistuje záruka, že se kód zkompile korektně u obou aplikací. Existuje možnost kompilace SWF souboru, který obsahuje pouze kód ActionScriptu ve Flash Builderu a poté nahrání do projektu vytvořeného v CS3. Stejně tak lze SWF soubor zkompilovaný v CS3 nahrát do projektu Flexu, ovšem jakmile se v těchto případech programátor odchýlí od jádra jazyka, začnou vyvstávat problémy.

Například Flex nemá takové grafické zdroje a komponenty jako má CS3 prostředí. Jde například o filmové klipy. Na druhou stranu CS3 nepodporuje tag pro vložení těchto komponent, který zase používá Flex. To znamená, že stejný kód, s těmito komponentami, nemůže být beztestně použit v obou případech. Stejně tak se liší architektura komponent, včetně jiného formátu komponent a celých jejich sad, které si neodpovídají.

Tyto problémy způsobily v komunitě diskuze a Adobe podniklo kroky k jejich nápravě a zlepšení kompatibility komponent. Avšak bude to asi ještě chvíli trvat, než bude možná pohodlná výměna kódu mezi aplikacemi a vývojovými balíčky. V tomto ohledu je na tom zatím nejlépe rozhraní AIR, které přímo vychází z CS3, takže tam zdaleka tolik problémů nevzniká. Znalost ActionScriptu tedy programátorovi značně

usnadní pohyb celou platformou v případě, že mu nebude stačit jen část, ovšem je třeba mít na paměti, že se problémy mohou vyskytnout a připravit se na ně.

3. ActionScript a jeho vývoj

V této druhé části se budu zabývat historií. Celá platforma Flash se již nějakou dobu vyvíjí a za tuto dobu nastalo spoustu změn, které stojí za povšimnutí. Nejprve zde upřesním průběžný vývoj tří hlavních částí platformy a to je přehrávač, vývojové prostředí a konečně jazyk ActionScript. Tyto tři části dám do souvislostí mezi sebou v rámci určité časové osy a v tom, jak spolu jednotlivé verze souvisely a přicházely na svět ve stejném čase.

Na konci se pak budu věnovat rozdílům mezi verzemi ActionScriptu a zaměřím se na verze 1.0 a 2.0. Verzi třetí se věnuje celá tato práce, a jelikož se jí bude z programátorského hlediska věnovat celá třetí část, nebudu jí zde rozebírat. Z další části budou dobře patrné rozdíly a inovace oproti verzím předchozím a budou probrané podrobněji.

3.1. Historie a vývoj platformy Flash

Základním problémem při psaní této kapitoly byl způsob rozdělení informací v ucelené celky, které spolu souvisejí. Nakonec jsem však došel k názoru, že tam kde to půjde, je vhodné rozdělit vývojové celky platformy pomocí verzí přehrávače Flash Player. Téměř každá vývojová etapa celé platformy totiž přišla s novou verzí, která musela podporovat nové vnitřní změny.

3.1.1. První kroky

Cesta k vývoji platformy Flash, tak jak ji známe dnes, začala v roce 1996, když malá softwarová firma FutureWave Software vydala již svůj druhý produkt s názvem FutureSplash Animator. Jejich prvním produktem byla aplikace SmartSketch, což byl vektorový grafický program určený pro počítače ovládané perem, který byl po rozmachu operačních systémů ovládaných myší importován jako do systému Microsoft Windows, tak do konkurenčního MacOS od firmy Apple. Po tomto úspěchu si firma uvědomila, že by měla přidat do programu tvorbu animací a použít jeho možnosti na tehdy velice bouřlivě se rozvíjejícím World Wide Webu, neboli aplikovat ho na

webových stránkách na internetu. Jedinou možností v té době bylo využití softwarové platformy Java, která byla soukromá a mnoha uživatelům internetu proto nevyhovovala. Tato situace se rychle změnila v okamžiku, kdy internetový prohlížeč Netscape³⁷ zavedl architekturu pluginů podobnou té, jakou známe z většiny prohlížečů dnes. Tato inovace umožnila zamýšlené vylepšení SmartSketchu a v květnu roku 1996 tedy vznikl FutureSplash Animator.

Technologie byla uplatněna na několika webech známých společností jako MSN³⁸ od Microsoftu, oficiální stránce seriálu Simpsonovi od společnosti Fox až po stránku společnosti Disney. Tato poslední zmíněná stránka také používala přehrávač Shockwave³⁹ od společnosti Macromedia, což umožnilo spojení s touto firmou. Macromedia později navrhla společnosti FutureWave, čítající v té době 6 lidí, spolupráci a v prosinci roku 1996 tuto firmu také odkoupila. Tímto aktem se z aplikace FutureSplash oficiálně stává vývojové prostředí Macromedia Flash 1. Změna to byla sice pouze kosmetická, ovšem právě tato změna započala cestu vývoje celé platformy Flash.

3.1.2. Macromedia Flash Player 2

Jedná se o první verzi vyvinutou společností Macromedia, která vyšla v roce 1997. Vývojový nástroj, který s touto verzí vyšel, se jmenoval Macromedia Flash 2 a až do verze 5 se toto číslování zachovalo, proto ho nebudu dále zmiňovat, dokud nenastane změna ve zvyku.

³⁷ Netscape je název série webových prohlížečů originálně vyvinutých korporací Netscape Communications. První verze tohoto prohlížeče byly velice oblíbené a dokonce dominantní na trhu. Avšak jako následek „první války prohlížečů“ ztratil skoro všechny své uživatele kvůli konkurenčnímu Internet Exploreru. Jeho vývoj skončil v roce 2008.

³⁸ The Microsoft Network – je kolekce internetových stránek poskytující informace o produktech společnosti Microsoft.

³⁹ Multimediální přehrávač vyvinutý společností Macromedia, který později přešel do vlastnictví společnosti Adobe. Specializuje se na přehrávání filmů a aplikací vytvořených aplikací Director (též od stejných firem).

Co se týká vylepšení, které tato verze přináší, není jich málo. Její hlavní funkce zůstává a je vylepšena vektorové grafika a pohyb. Toto však není tak revoluční změna jako je přidání zvuku do animací. Druhá verze přehrávače již podporuje základní stereo zvuk, čím se přehrávač stává již více multimediální. Další změnou byla možnost používat v animacích bitmapy, a proto se již nejedná o striktně vektorový přehrávač. Poslední grafickou úpravou bylo přidání funkce tween⁴⁰.

Při vývoji animací pro tento přehrávač se také vývojáři dočkali určitých vylepšení. První je možnost přidat do animace tlačítka a druhá je knihovna objektů, použitelná při práci s vývojovým nástrojem. S tlačítky také souvisí vylepšení skriptů. Přehrávač verze 2 totiž byl první, který měl podporu skriptování a obsahoval tedy jakéhosi prapředka ActionScriptu. Akcí, které byl schopen provádět nebylo mnoho, ale i tak pomohly vývojářům v jejich práci. Akce obsahovaly například příkazy gotoAndPlay (skoč a přehraj), gotoAndStop (skoč a zastav), nextFrame (další snímek) a nextScene (další scéna), které umožňovaly práci s časovou osou.

3.1.3. Macromedia Flash Player 3

V roce 1998 vyšla další verze přehrávače spolu s vývojovým prostředím. Z hlediska přehrávače v této verzi bylo přidáno několik věcí. Jedním z vylepšení animací bylo přidání alfa průhlednosti⁴¹, která obohatila grafické portfolio funkcí. Dalším vylepšením přehrávače, kromě zlepšené možnosti publikace na internetu, bylo přidání možnosti komprimovat zvuk ve formátu MP3, čímž se znatelně snížila velikost animací se zvukem.

⁴⁰ Tweening je grafický proces generování snímků mezi dvěma obrázky tak, aby to vypadalo, že první přešel plynule do druhého.

⁴¹ Průhlednost nebo také transparentnost umožňuje v aplikacích při překrytí obrázků nechat určitá místa průhledná a zobrazit tak obrázek umístěný vespod. V rastrových obrázcích je toto docíleno buď pomocí nastavení průhledné barvy anebo pomocí alfa kanálu.

Vývojáři s nástrojem verze 3 mohli například nově použít element movieclip⁴², který se využívá dodnes. Dalším zlepšením vývojového nástroje byl plugin umožňující využít tehdejší verzi jazyka JavaScript a v neposlední řadě pak dostali možnost vytvářet samospustitelné animace, k jejichž přehrání nepotřeboval uživatel mít v počítači nainstalovaný Flash Player. Z hlediska samotného skriptování mnoho změn nenastalo, snad jen k funkcím zmíněným v předchozí kapitole přibýlo několik dalších podporujících interaktivitu a příkaz loadMovie (načti film), umožňující načíst externí SWF soubor a dále s ním pak pracovat.

3.1.4. Macromedia Flash Player 4

Spatřil světlo světa v květnu roku 1999 a ohromil uživatele internetu možností streamování hudby ve formátu MP3, což dosud nebylo možné. Toho se chytilo hned několik společností nabízejících hudbu k prodeji na internetu a využilo tuto funkci k přehrávání ukázek děl na prodej. Druhým zlepšením, které uživatelé mohli rovnou „vidět“, byl tzv. Motion Tween, tedy stejná technologie používaná u obrázků, ovšem v tomto případě použitá pro volný pohyb objektu z místa na místo pomocí časové linie. Poslední větší inovací bylo přidání základní formulářové komponenty input, umožňující zadat uživateli do Flash aplikace určitá data.

Na první pohled toho tedy zase tolik nebylo, ovšem hlavní změny nastaly uvnitř. Tato verze Flashe totiž přišla s plnou podporou skriptů, které se nazývaly Actions, neboli česky akce. Skriptovalo se pomocí již známé syntaxe Flashe a tyto skripty už obsahovaly mnoho funkcí vyššího programovacího jazyka. Umožňovaly například cykly, podmínky, práci s proměnnými a jiné základní programátorské konstrukty. Vývojové prostředí těmto změnám bylo uzpůsobeno a obsahovalo již tedy základní funkce běžně používané v jiných programovacích jazycích.

Zajímavou informací z této doby je, že Flash Player plugin původně nebyl distribuován automaticky s prohlížečem, ale uživatel se musel dostavit na stránku firmy Macromedia a plugin stáhnout a nainstalovat. To se změnilo v roce 2000, kdy byl Flash

⁴² Movieclip je element používaný při tvorbě souborů platformy Flash. Jedná se v podstatě o kontejnerový element, který může obsahovat jiné elementy nebo animace. Jednoduše řečeno se jedná o animaci uvnitř animace.

Player již běžnou součástí prohlížečů Netscape a Internet Explorer. Dva roky na to už základní instalace putovala k zákazníkům jako součást operačního systému Windows XP. V té době se podle statistik tato základní instalace dostala k 92 procentům uživatelů internetu.

3.1.5. Macromedia Flash Player 5

V červenci roku 2000 přišli vývojáři z firmy Macromedia s novou verzí Flashe, která oproti té předchozí poskočila výrazně kupředu, co se týče schopností skriptů. V této verzi totiž byly opět značně posíleny již známé Actions z předchozích verzí a byly poprvé oficiálně pojmenovány ActionScript. Tato první verze skriptovacího jazyka používala prototypové programování⁴³ a byla založena na ECMA skriptu, o kterém budu podrobněji mluvit v další části této práce. ActionScript v této verzi již dovoľoval plně, jak objektově orientované⁴⁴, tak procedurální⁴⁵ programování. ActionScript verze 1.0 proberu více na konci této části, abych ucelil jeho porovnání s dalšími verzemi. Pro účely této kapitoly zatím tato krátká informace postačuje.

Kromě lepšího skriptovacího jazyka tato verze Flashe doznala zlepšení hlavně ve svém vývojovém nástroji. Macromedia Flash 5 byl poprvé schopen řádně nastavit svůj interface tak, jak si uživatel představoval a usnadnil mu tak práci s ním. Co se týká obsahu nástroje, byla také poprvé implementována obsluha XML dokumentů. Dále byly přidány tzv. Smartclips, což byly předchůdci dnešních programovacích komponent. Poslední velkou změnou pak bylo přidání možnosti HTML formátování do dynamického textu ve všech komponentách. To umožnilo snadnější formátování textu ve Flashovém okně a lepší integraci Flashe do struktury webové stránky.

⁴³ Programování založené na prototypch je druh objektově orientovaného programování, ve kterém nejsou přítomny třídy a proces dědičnosti je prováděn klonováním existujících objektů sloužících jako prototypy.

⁴⁴ Objektově orientované programování neboli OOP je styl programování využívající objekty, což jsou datové struktury obsahující atributy a metody, a jejich vzájemnou interakci k tvorbě aplikací.

⁴⁵ Procedurální programování je druh programování odvozený od strukturovaného programování. Využívá se zde volání procedur a funkcí jako částí subrutin a podprogramů.

3.1.6. Macromedia Flash Player 6

Verze 6 již v té době oblíbeného přehrávače vyšla v květnu roku 2002 pod kódovým označením Exorcist (Vymítač). Bylo to poprvé, co číslo u přehrávače nesouhlasilo s názvem vývojového nástroje, protože tentokrát nástroj dostal označení Macromedia MX.

Nové schopnosti přehrávače byly každému uživateli internetu na první pohled zřejmé, protože verze 6 byla první verzí podporující streamování audio/video živě a nebo případně na požádání. Toho bylo dosaženo implementací protokolu a RTMP⁴⁶ a serverovou aplikací Macromedia Flash Communication Server MX, která vyšla rovněž v roce 2002 a umožňovala streamovat video přímo do Flash Playeru. Bez ní mohlo být video pouze ve Flash souboru zakomponováno, ovšem nemohlo být vyžádáno online. A dokonce ani to nebylo předtím možné a dosaženo toho bylo implementací video kodeku Sorenson Spark , o kterém se zmiňuji v první části.

Další výhodou nového přehrávače byla možnost posílat data mezi jednotlivými Flashovými a nebo webovými aplikacemi pomocí formátu AMF⁴⁷ a protokolu SOAP⁴⁸. Tímto bylo možné u Flashových webových aplikací docílit synchronizace dat anebo výměny informací, čehož bylo využíváno nejenom ve hrách. Posledním velkým vylepšením, co se přehrávače a toho co uživatel na první pohled uvidí týče, byla podpora tzv. screen readerů. To jsou aplikace určené pro zrakově postižené, které se snaží interpretovat to, co je zobrazeno na obrazovce, a převádí to na čtený text, zvukové signály či hardware převodník na braillovo písmo. Konkrétně toho bylo docíleno přes platformu MSAA⁴⁹, která je pro takové aplikace určená.

⁴⁶ Real Time Messaging Protocol je protokol vlastněný firmou Adobe a vyvinutý firmou Macromedia a slouží ke streamování audia, videa a různých dat po internetu.

⁴⁷ Action Message Format je binární formát určen k uchování dynamických objektů v ActionScriptu. Primárně je používán k výměně dat mezi aplikacemi Adobe a vzdálenými službami, hlavně přes internet.

⁴⁸ Simple Object Access Protocol je specifikace protokolů pro výměnu strukturovaných informací přes vzdálené služby a počítačové sítě.

⁴⁹ Microsoft Active Accessibility

Vývojové prostředí doznalo také určitých změn a to hlavně v usnadnění práce vývojářům. Prostředí zavedlo kódování Unicode, čímž dovolilo do programového kódu a textu zadávat více druhů znaků a kromě zlepšení komprese pracovních souborů a již zmíněné podpory videa se toho už moc nezměnilo. Co se týká jazyka, na druhou stranu pár změn logicky nastat muselo. Jelikož se celá tato verze točila kolem streamování videa, byly do jazyka přidány obslužné funkce a také celá řada komponentů pro uživatelské rozhraní. Důležitý pak byl soubor funkcí pro vykreslování vektorové grafiky.

3.1.7. Macromedia Flash Player 7

V září roku 2003 vyšla další stěžejní verze přehrávače, tentokrát pod jménem Mojo. I tato verze doznala změn ve všech třech hlavních směrech. Přehrávač jako takový získal schopnost tvořit lépe přehledy, formuláře, tabulky a grafy a zároveň se mu zlepšilo streamování videa, které bylo od té doby prováděno pomocí protokolu HTTP. Další novou funkcí, i když nebyla vyhraněna vyloženě na Flash Player, byla možnost importovat soubory do formátu PDF⁵⁰ a do aplikace Adobe Illustrator⁵¹. Jelikož formát i Aplikace byly vyvinuty firmou Adobe Systems, začíná tím také externí spolupráce s touto firmou.

Nejzajímavější hledisko pro tuto práci bylo všem zavedení ActionScriptu 2.0. Tato verze je již plně objektově orientovaná a obsahuje tedy třídy a používá klasickou dědičnost namísto prototypů, jak tomu bylo v předchozím případě. Díky tomuto byli vývojáři nuceni kompletně přepsat komponenty z předchozí verze nástroje MX, tak aby vyhovovaly objektově orientovanému přístupu. Více o této konkrétní verzi jazyka budu mluvit na konci této části kvůli srovnání s jinými verzemi. Jako zajímavost, ale uvedu, že ač byly SWF soubory s touto verzí jazyka nepoužitelné pro přehrávače předchozí, existovala možnost překompilovat tuto verzi tak, aby vznikl kód, který pro přehrávač verze 6 vypadal, že byl napsán v ActionScriptu 1.0 a tím pádem byl pro tuto verzi rozpoznatelný a použitelný.

⁵⁰ Portable Dokument Format je datový formát vyvinutý firmou Adobe systems v roce 1993 určený k výměně a zobrazování dokumentů.

⁵¹ Je vektorový grafický editor vyvíjený firmou Adobe Systems.

Díky inovaci jazyka bylo nutné upravit také vývojový nástroj. Tento se jmenoval Macromedia Flash MX 2004 a vyšel ve dvou verzích. Verze Professional obsahovala kromě vylepšení jazyka z normální verze také více nových komponent, integraci webových služeb a pro uživatele pohodlnější nástroje pro práci s videem a audiem. Kdo si připlatil, dostal s lepší verzí také možnost zakomponovat do SWF souborů přehrávače Flash videa a nebo MP3 hudby. Ačkoliv verze Professional obsahovala mnoho oproti verzi základní, ani jedna verze však neměla pomocníka při psaní kódu, který je běžný pro všechny vyspělé programovací nástroje a dokonce byl běžný i u předchozí verze tohoto jazyka. Díky tomu byl nucen vývojář všechny kód psát ručně a nemohl se spoléhat na chytré a rychlé doplňování kódu programem.

V roce 2004 byla poprvé oficiálně představena platforma Flash. To bylo umožněno expanzí Flashe i do jiných vývojářských nástrojů a ne jenom těch, které jsem doposud zmiňoval. V souvislosti s tímto byly zavedeny dva programy. První byl Flex a jak jsem již zmínil výše v poznámce, byl to program pro tvorbu bohatých internetových aplikací. Druhým byl Breeze a ten se specializoval na tvorbu prezentací, online materiálů, webové konference a e-learning. Obě tyto aplikace využívaly Flash Player jako formu doručení svého produktu zákazníkům, ovšem používaly k jeho vytvoření jiný vývojový nástroj, než byl Macromedia Flash. Obě tyto aplikace vyvinula firma Macromedia a obě po jejím převzetí nadále vyvíjela firma Adobe.

3.1.8. Macromedia Flash Player 8

Pod názvem Maelstorm (Bouře) vyšla v červenci roku 2005 poslední verze přehrávače vydaná firmou Macromedia. V této verzi se uživatelé internetu dočkali opět nových schopností počínaje dynamickým nahráváním obrázků v rastrových formátech GIF⁵² a PNG⁵³. Jelikož jsou rastrové obrázky větší, než vektorové, byli vývojáři přehrávači nuceni zlepšit výkonnost přehrávače a přidat kešování rastrových obrázků. To také souviselo s novou schopností přehrávače přímo stahovat a nahrávat soubory z a

⁵² Grafic Interchange Format je grafický formát určený pro rastrovou grafiku. Používá bezztrátovou kompresi LZW84.

⁵³ Portable Network Graphics, což česky znamená přenosná síťová grafika, je grafický formát určený pro bezztrátovou kompresi rastrové grafiky a byl vyvinut, aby nahradil formát GIF.

do uživatelského počítače. Z hlediska videa pak došlo k zavedení nového kodeku, kterým byl již dříve zmíněný VP6. Poslední novou funkcí, která stojí za zmínku, je přidání dynamických filtrů, kterými šly způsobovat grafické efekty jako je rozmazání obrazu a zobrazení stínů za objekty.

Z hlediska ActionScriptu tentokrát nedošlo k výraznějšímu zlepšení. Jediné, co stojí za zmínku, je přidání enginu pro lepší vykreslování textu a knihovny funkcí pro práci s rastrovou grafikou a funkcí, které jsou již zmíněny výše. Zajímavá je ovšem u této verze profilace vývojových nástrojů, které tentokrát opět byly dva. Macromedia Flash 8 Basic byla základní verze, která se snažila přilákat nové uživatele, kteří chtěli pouze kreslit, vytvářet animace a pracovat s vektorovou grafikou. Tato verze má pouze omezenou podporu videa a lepších animačních efektů. Oproti tomu verze Professional byla zaměřena na všechny možnosti vyjádření, kvalitu a hlavně video. Obsahovala všechny vymoženosti, které jsem vyjmenoval výše a které ve verzi Basic nebyly dostupné.

Dne 3. prosince 2005 firma Adobe Systems získala firmu Macromedia a celé její portfolio produktů, což znamenalo i platformu Flash. Některé z nich, a Flash nevyjímaje, firma nadále vyvíjela a činí tak dodnes.

3.1.9. Adobe Flash Player 9

Již pod novým jménem vyšel pod kódovým označením Zaphod v červnu 2006 Adobe Flash Player 8.5, který byl později přejmenován na verzi 9. V této verzi firma Adobe představila programovací jazyk ActionScript 3.0, který ve víceméně nezměněné verzi používá dodnes. To přineslo obrovskou změnu ve vývoji Flashových aplikací. Více se tímto jazykem budu zabývat dále. Zatím pouze uvedu, že přinesl jak nové možnosti, tak hlavně zlepšení výkonu samotných aplikací. Ač tato verze představila nový jazyk, ActionScript 2.0 byl stále aktuální a tato verze pro něj měla podporu a byla schopná jak používat starší Flashové soubory, tak je dále vyvíjet v aktuálním vývojovém prostředí.

Kromě uvedení nového jazyka se Flash dočkal také drobných změn ve funkčnosti. Prvně to byla podpora regulárních výrazů⁵⁴ a jmenných prostorů⁵⁵, což usnadnilo a otevřelo nové možnosti v programování aplikací. Zvýšení výkonnosti bylo hlavním tématem této verze a k tomu kromě jazyka přispěl také nový JIT⁵⁶ kompilér. Poslední velkou změnou byl nový způsob zpracování XML souborů a to pomocí metody EX4⁵⁷.

K vývojovému nástroji firma Adobe přistoupila trochu jinak než firma Macromedia. Vydala ho totiž v rámci balíků svých produktů CS3 Suite. Jmenoval se Adobe Flash CS3 Professional a kromě již zmíněných výhod, které přinášel jazyk ActionScript, zaměřoval se hlavně na integraci Flashe s jinými produkty firmy Adobe jako byl Adobe Photoshop a nebo Adobe Illustrator.

Adobe Flash Player 9 updates

Adobe se rozhodlo při minimálních vylepšeních přehrávače a jeho schopností nevydávat další nové verze, ale pouze upgradovat stávající. Verze 9 má celkem tři upgrady, které vyšly v rozmezí jednoho roku.

První vyšel v listopadu roku 2006 a měl označení Marvin. Přinášel v podstatě jen jednu pro uživatele podstatnou věc a to byla podpora módu fullscreen, což je označení, které se používá v okamžiku, kdy je aplikace roztažena přes celou obrazovku a není vidět nic jiného než její rozhraní.

Druhý vyšel v červnu 2007 pod kódovým jménem Hotblack a obsahoval v podstatě jen bezpečnostní opravy, na které bylo Adobe upozorňováno už delší dobu. Nejdůležitějším upgradem verze 9 byl upgrade číslo 3. Ten vyšel v prosinci 2007 pod

⁵⁴ V počítačové terminologii se pojmem „regulární výraz“ označují prostředky používané k porovnávání a vyhledávání řetězců textu. Například se jimi nahrazují písmena, slova nebo celé fráze.

⁵⁵ Je mechanismus v programování, sloužící ke sdružení logických celků, které spolu souvisejí.

⁵⁶ Just-in-time kompilace je druh kompilace často také nazývaný dynamický překlad. Jde o techniku k vylepšení výkonnosti běžícího počítačového programu.

⁵⁷ ECMAScript for XML je vylepšení programovacích jazyků založených na ECMA skriptu, které přináší nativní podporu pro dokumenty jazyka XML.

přezdívkou Moviestar (Filmová hvězda). Toto označení bylo vsutku na místě, protože se všechny jeho vylepšení týkaly filmu a videa.

Jednalo se zejména o zavedení video a audio kodeků H.264 a AAC. Tyto kodeky jsou špičkou audiovizuální komprese dodnes a Adobe jejich zavedením nejen zlepšilo kompresi Flash videa, ale také přistoupilo na požadavky trhu. Další podporou videa bylo zavedení formátu F4V, který stál na normě ISO pro základní mediální formáty a kontejnery⁵⁸. Tento konkrétní formát byl MPEG-4⁵⁹ část 12.

3.1.10. Adobe Flash Player 10

V říjnu roku 2008 vyšla zatím poslední verze a to verze desátá pod označením Astro. Stejně jako tomu bylo v předchozích případech, došlo k přidání několika nových schopností a také k posílení několika již předtím zavedených. V případě vylepšených to byla už poněkolidáté zlepšená podpora rastrových obrázků a lepší soubor funkcí pro vykreslování. Dalším zlepšením bylo nové kontextové menu pro aplikace. Posledním větším zlepšením již existujícího principu bylo dodání Speex audio kodeku pro lepší kompresi mluveného zvuku, o kterém jsem se zmiňoval již v první části.

Co však v této verzi přibýlo nového a na co se hlavně specializuje, je práce se 3D objekty a jejich různé transformace. Díky náročnosti těchto úkonů byla také zlepšena hardwarová akceleraace grafiky a byl přidán nový engine podporující anti-aliasing⁶⁰. Jelikož práce s 3D grafikou je pro procesor náročná, byly některé procesy převedeny na GPU, tedy na výpočetní jednotku grafické karty, která je k takovým účelům uzpůsobena a bylo tak docíleno zobrazení více snímků za sekundu. Další novinkou bylo dynamické generování zvuků a lepší vykreslování a druhy textů, se kterými se dá pracovat.

⁵⁸ Tato norma definuje obecnou podobu multimediálních formátů závislých na čase jako je video a audio.

⁵⁹ Je multimediální kontejnerový formát nejčastěji používaný k ukládání digitálních audio a video streamů.

⁶⁰ Jedná se o grafický proces, kterým se docílí vyhlazování hran u 3D objektů.

Poslední velkou novinkou bylo přidání Peer-to-Peer⁶¹ komunikace mezi klienty pomocí protokolu RTMFP⁶², čehož bylo využíváno hlavně v online hrách anebo aplikacích využívajících komunikaci mezi uživateli.

Hlavní vývojový nástroj určený pro Flash následoval svého předchůdce ve stylu pojmenování a protože se tentokrát za Adobe Creative Suite přidalo číslo 4, tak se také za Adobe Flash přidala zkratka CS4. Změny nastaly hlavně v designu a rozmístění nástrojů a to právě kvůli přidání práce s 3D objekty. Do ActionScriptu se přidala podpora nových funkcí, ale jinak se jeho struktura nějak zásadně neměnila.

3.2. Srovnání starších verzí ActionScriptu

Ke konci této části bych rád poskytl informace o dvou starších verzích ActionScriptu, které jsem již zmínil v předchozích kapitolách. S těmito verzemi už se uživatel tak běžně nesetká, i když především verze 2.0 se pořád ještě vyskytuje a někteří programátoři ji používají kvůli neochotě přestupu na verzi 3.0, jež prošla značnou syntaktickou změnou.

3.2.1. ActionScript 1.0

S vypuštěním Flash Playeru 5 v září 2000 nastalo vylepšení již známých „akcí“ z předchozích verzí do takové míry, že se poprvé mluvilo o samostatném jazyce. Tento tehdy čistě skriptovací jazyk byl nazván ActionScript 1.0 a používal se až do roku 2003. Byla to také první verze ActionScriptu, kterou ovlivnil jazyk JavaScript a byla postavená na normě ECMA-262, což je třetí generace této normy. Díky tomu podporovala tato verze objektový model této normy a mnoho z jejích klíčových datových typů.

Lokální proměnné mohly být v této verzi definované pomocí klíčového slova `var` a uživatelsky definované funkce s návratovým parametrem bylo také možné nově

⁶¹ Neboli komunikace rovný s rovným je v počítačové terminologii komunikace mezi dvěma klientskými aplikacemi bez podpory serverové strany.

⁶² Real Time Media Float Protocol je protokol vyvinutý firmou Adobe a slouží k peer-to-peer komunikaci mezi aplikacemi stvořenými ve Flashi.

vytvářet. Oproti předchozí „verzi“ mohl být také ActionScript vytvářen přímo v textovém editoru místo toho, aby byl uživatel nucen skládat dohromady akce z výběrového menu a dialogových oken.

S vypuštěním další verze vývojového nástroje Flash MX zůstal jazyk v podstatě nezměněn. Vzniklo v něm pouze několik drobných úprav jako zavedení konstrukce switch na větvení podmínek a také zavedení operátoru pro striktní rovnost ===. Obě tyto změny jazyk přinesly opět blíže k normě ECMA-262.

To co ovšem jazyk zásadně odlišovalo od následujících verzí byl volný typový systém a již zmiňovaná prototypová dědičnost. Volné typování označuje schopnost proměnných udržovat v sobě jakýkoliv typ dat. Tento systém zaručuje rychlý vývoj a psaní skriptů a je velmi vhodný pro skriptování na malých projektech. Dědičnost na bázi prototypů je mechanismus, který zaručuje ActionScriptu 1.0 schopnost opakovaně používat stejný kód a dává mu možnost velmi omezeného objektově orientovaného programování. Místo klíčového slova class použitého pro třídu, se v této verzi nachází speciální objektový typ, který slouží jako prototyp pro třídu objektů. Všechny běžné charakteristiky třídy jsou definovány v prototypu této třídy a každá instance třídy obsahuje ukazatel na základní prototyp.

3.2.2.ActionScript 2.0

Další hlavní revizí jazyka byla verze 2.0, která byla představena v září 2003, tedy přibližně 3 roky po té první. S ní vyšel nový vývojový nástroj Flash MX 2004 a odpovídající přehrávač verze 7. Tato verze byla odezvou na požadavky uživatelů, aby byl ActionScript více vybaven pro tvorbu robustnějších a komplexních aplikací. Kvůli těmto požadavkům byl zaveden systém kontroly datových typů při kompilaci a také syntaxe na bázi tříd jako je tomu u většiny objektově orientovaných jazyků.

Tyto dvě zásadní úpravy dovolovaly jazyku více objektově orientovaný přístup, ovšem kód se stále dal zkompileovat tak, aby odpovídal verzi 1.0 a dovoloval tak spuštění ve Flash Playeru verze 6. Jinými slovy z toho plyne, že dědičnost na bázi tříd, kterou tento jazyk zavedl, byla pouhou nadstavbou prototypového programování z předchozí verze. Nicméně tato nadstavba umožnila uživatelům tvořit své vlastní třídy a rozhraní tak, jak tomu byli zvyklí z objektově orientovaných jazyků jako Java nebo

C++. A navíc byla oproti prototypovému přístupu mnohem přehlednější, čímž usnadnila programování větších projektů.

S ActionScriptem 2.0 mohli vývojáři poprvé omezit proměnné na jeden datový typ a kompilátor je pak upozornil na problémy, které v kódu nastávají, pokud udělali chybu a přiřadili nevhodné hodnoty do nevhodné proměnné. Což značně usnadnilo opravu chyb, které ve verzi 1.0 často nastávaly. Tato verze také už částečně odpovídala standardu ECMA čtvrté generace.

4. Rozbor ActionScriptu 3.0

Ačkoliv nová verze interního jazyka platformy Flash obsahuje mnoho toho, s čím jsou uživatelé předchozích verzí již seznámeni, bude nejspíš nejlepší přemýšlet o ActionScriptu 3.0 jako o úplně novém programovacím jazyce. Toto má několik důvodů. Za prvé několik věcí je v tomto jazyce zcela nových, jako třeba systém událostí anebo způsob, jakým jsou zobrazovány komponenty. Za druhé drobné změny procházejí celým jazykem a chvíli trvá, než si na ně programátor zvykne. Jsou to například lehké změny názvů atributů a metod.

Nicméně největší důvod pro to považovat ActionScript 3.0 za úplně nový je ten, že tento jazyk byl úplně přepsán a používá tedy úplně jinou kódovou základnu než jeho předchozí verze. Tato optimalizace přináší jazyku relativně dramatické zvýšení výkonu, avšak zapříčiňuje také, že tato verze nemůže být smíchána s verzemi předchozími a je tudíž s nimi nekompatibilní a nemůže se s nimi proto vyskytovat ve stejném souboru.

Novota této verze by neměla programátora zastráhit. Je pravda, že učební křivka tohoto jazyka je strmější než u předchozích verzí, ale to je zapříčiněno spíše robustností jazyka než složitostí. Nicméně je pravda, že při učení tohoto jazyka musí programátor znatý předchozích verzí projít určitým obdobím kdy si zvyká na trochu jiný způsob, jakým jsou zde věci prováděny.

4.1. Účel a prostředky

Hlavní motivací k vývoji ActionScriptu 3.0 bylo stvořit jazyk ideální pro velmi rychle se rozvíjející odvětví bohatých internetových aplikací, které se poslední dobou stávají neodmyslitelnou součástí webu. Dřívější verze ActionScriptu nabízely potřebné prostředky a flexibilitu k tvorbě opravdu zajímavých webových aplikací. ActionScriptu 3.0 teď toto posouvá ještě dál a poskytuje bezvadnou výkonnost a relativně jednoduchou obsluhu k tvorbě vysoce komplexních aplikací použitelných v různých odvětvích. A pokud vývojářům nevadí, že uživatel musí ve většině případů ke spuštění jejich výtvorů použít Flash Player, mohou se svou aplikací dosáhnout excelentní produktivity a výkonu.

4.1.1. ECMA skript

ECMA skript je skriptovací jazyk standardizovaný firmou Ecma International, což je nezisková soukromá organizace pro standardy. Tento jazyk je velice rozšířen na webu hlavně díky svým třem nejznámějším dialektům a to JavaScript, ActionScript a JScript.

Standard vznikl tak, že po představení JavaScriptu firmou Netscape v roce 1995 a jeho následném velkém úspěchu mezi vývojáři dynamických webových stránek Microsoft vyvinul dialekt tohoto jazyka s názvem JScript. Ovšem díky tomu, že v podstatě jeden podobný jazyk vyvíjely dvě firmy, vznikaly mezi jazyky velké nekonzistence. Aby se tomu zabránilo, firma Netscape na konci roku 1996 dodala JavaScript firmě Ecma International pro standardizaci. Díky tomu vznikl standard ECMA-262, který má doteď už 5 generací.

ActionScript 3.0 je dialektem právě tohoto standardu ECMA-262 a ač je perfektně kompatibilní s jeho třetí generací, jeho funkčnost stojí spíše na generaci čtvrté.

4.1.2. Virtuální stroj

Virtuální stroj je software, který vytváří virtualizované prostředí mezi platformou počítače a operačním systémem. Díky němu může uživatel provozovat specifický software na abstraktním stroji.

ActionScript je spouštěn pomocí takzvaného ActionScript Virtual Machina neboli AMV. To je virtuální stroj zabudovaný přímo do Flash Playeru, který v podstatě umožňuje spouštět aplikace stvořené na platformě Flash. Než vyšel ActionScript 3.0, všechny aplikace na platformě Flash obsluhoval virtuální stroj AMV1. Nicméně vývojáři začali postupně tlačit tento stroj k jeho výkonnostním limitům, a proto s nástupem nové verze jazyka nastupuje i nový virtuální stroj s názvem AMV2. Tento stroj pak dramaticky navyšuje výkon v porovnání s AMV1. Jako důsledek této změny se kód napsaný pomocí ActionScriptu 3.0 vykonává až desetkrát rychleji, než tomu bylo u jeho starších verzí.

Nová verze virtuálního stroje je dostupná ve Flash Playeru 9 a bude primárním virtuálním strojem pro spouštění aktuálního kódu a kódu, který bude následovat.

Nicméně pro zpětnou kompatibilitu bude do nových verzí Flash Playeru dodáván i AVM1, aby bylo možné spouštět i obsah vytvořený pomocí starších verzí jazyka.

4.2. Cíle jazyka

Po vydání jazyka vývojáři představili cíle, které měli na paměti při tvorbě jazyka. V první řadě jim šlo o to doručit uživatelům vylepšený programový model, který bude v souladu s moderními standardy a výkonnostně na mnohem vyšší úrovni, než představili ve svých předchozích verzích. Aby toho dosáhli, použili sice nový programový model, ale je to takový model, který bude již známý programátorům se základní znalostí objektově orientovaného programování.

K dosažení těchto cílů se pak při vývoji drželi několika jednoduchých bodů.

- **Bezpečnost:** Jazyk podporuje typovou bezpečnost, takže vývojář může psát jednoduchý a snadno zvládnutelný kód, který je přehledný a zároveň ho díky tomu kompilátor upozorní na chyby spojené s typem proměnných.
- **Jednoduchost:** Jazyk je dostatečně jednoduchý a intuitivní na to, aby byli vývojáři schopni programy číst bez neustálého nahlížení do manuálů.
- **Výkon:** Jazyk dovoluje psát komplexní programy, které mají efektivní výkon a na uživatelské pokyny reagují bez problémů.
- **Kompatibilita:** Jazyk sice není přímo zpětně kompatibilní s předešlou verzí, ale vychází ze stejného standardu ECMA, a proto se dá vycházet nejen z předešlých verzí, ale také z jiných dialektů tohoto standardu. Navíc přidává podporu ECMA pro XML (E4X), která je již dnes běžnou součástí ostatních kompatibilních dialektů.

4.3. Vlastnosti jazyka

ActionScript 3.0 sestává ze dvou částí. První je jádro jazyka a druhá je Flash Player API⁶³. Jádro programu definuje základní stavební bloky programovacího jazyka, jako jsou struktury, výrazy, podmínky, cykly a datové typy. Flash Player API je tvořeno třídami, které zajišťují přístup ke specifickým funkcím přehrávače. Tyto dvě části ještě doplňuje E4X, které je sice součástí jádra jazyka, ale svou funkcí a díky novým možnostem, které přináší, se dá postavit mimo něj.

4.3.1. Jádro jazyka

ActionScript 3.0 si bere jádro z ActionScriptu 2.0 a upravuje ho tak, aby plně vyhovovalo standardu ECMA. Po této úpravě mu však ještě dodává úplně nové oblasti funkčnosti a otevírá tak programátorovi nové možnosti. Všechny tyto možnosti a novinky jsou dopodrobna rozebrány v referenčním manuálu jazyka ActionScript 3.0 dostupného na stránkách Adobe. V této kapitole z nich vyberu ty nejdůležitější a popíši jejich funkci a význam.

Běhové výjimky

V ActionScriptu 2.0 mnoho běhových chyb shodilo aplikaci sice dokonale, nicméně velmi tiše. Toto zaručovalo že Flash Player nezobrazí uživateli pro něj nevysvětlitelné upozornění o neznámé chybě, jako to dělal JavaScript ve starších verzích webových prohlížečů. Na druhou stranu však tento nedostatek ohlašování chyb pak velice ztěžoval jejich opravu programátory.

ActionScript 3.0 představuje nový systém běhových výjimek, které pokrývají většinu běžných chybových situací. Díky tomuto systému je možné opravovat chyby mnohem snadněji a dokonce programy psát takovým způsobem, aby si s chybou poradily samy, když nastane. Tento systém pak je schopný dokonce určit soubor a místo v kódu, kde chyba nastala, čímž její opravu programátorovi neuvěřitelně zjednodušuje.

⁶³ Application Programming Interface je rozhraní implementované do softwarového programu, které jí umožňuje interakci s jiným softwarem. Zajišťuje interakci s jinými programy stejně, jako uživatelské rozhraní zabezpečuje interakci mezi lidmi a počítači.

System datových typů

V ActionScriptu 1.0 se proměnné netypovaly vůbec. Všechny byly stejné a programátor k nim mohl přiřadit jakýkoliv obsah. Toto působilo při omylu programátora značné problémy. V ActionScriptu 2.0 byly pak tyto problémy vyřešeny přidáním typů k proměnným. Tyto typy byli však hlavně pomoc pro programátora a pro kompilátor, který ho pak upozornil na případné nebezpečí. Za běhu programu se pak všechny typy u proměnných zase zapomněly.

Ve třetí verzi jazyka je toto dotaženo ještě trochu dále. Typové informace jsou za běhu zachovány a používají se několika způsoby. Například Flash Player provozuje pravidelné kontroly typů proměnných, které jsou spolu v interakci, a tím dále zlepšuje typovou bezpečnost. Typové informace jsou pak také použité, když se hodnota v proměnné promítá do místního nastavení počítače. A v neposlední řadě znalost typu proměnné za běhu umožňuje šetřit místo v paměti tím, že se jí vyhradí jen určitý prostor, přes který se hodnota nemůže dostat.

Zamčené třídy

Novinkou ve třetí verzi ActionScriptu je také koncept zamčených tříd. Zamčené třídy mají v době kompilace fixně nastavený počet metod a atributů a tento počet se nemůže měnit. Ve většině objektově orientovaných jazyků mohou být takové třídy vytvořeny pomocí klíčového slova `sealed`. Znamená to pak, že tuto třídu nelze upravovat a nelze od ní dědit další třídy.

V ActionScriptu není klíčové slovo potřeba, protože jsou defaultně uzamčeny všechny třídy. To způsobuje striktnější kontrolu při kompilaci a také to šetří paměť, protože si objekty dané třídy za běhu nemusí udržovat tabulku se seznamem metod a atributů. Samozřejmě jsou zde také možné dynamické třídy, které jsou opakem tříd zamčených, nicméně ty musí být přesně definovány pomocí klíčového slova `dynamic`.

Dynamické metody

Obsluha událostí je v ActionScriptu dramaticky zjednodušena díky novému způsobu dynamických metod, který zabezpečuje snadnou delegaci událostí. Dynamická metoda je taková metoda, která vznikne dynamicky až za chodu programu. V ActionScriptu 2.0 si taková funkce nepamatovala, z jakého objektu byla vytvořena,

což způsobovalo nepředvídatelné chování, když taková funkce byla zavolána. To bylo ve starší verzi jazyka obcházeno pomocí třídy `mx.utils.Delegate`, nicméně v nové verzi už toho není zapotřebí. Jakmile událost nastane, je vygenerována obslužná funkce, která si následně pamatuje, jakého objektu vznikla a který tedy obsluhuje.

4.3.2. ECMA pro XML

ActionScript 3.0 se může pochlubit plnou implementací standardu ECMA pro XML. Tento standard s označením ECMA-357 poskytuje přirozenou a rychlou cestu jak manipulovat s XML soubory. Na rozdíl od tradičních API k tomuto určených tento standard přidává XML dokument jako datový typ přímo do programovacího jazyka. Díky tomu drasticky snižuje množství kódu potřebných pro práci s XML dokumenty a tím pádem i celou operaci zjednodušuje

Regulární výrazy

ActionScript 3.0 má v sobě podporu regulárních výrazů, takže je velice snadné rychle vyhledávat a manipulovat řetězce textu. Regulární výrazy pracují jako filtry při vyhledávání a práci s textem. Výraz samotný slouží jako maska, která definuje určitá kritéria pro porovnávání textu. ActionScript 3.0 obsahuje regulární výrazy definované v ECMA standardu ECMA-262.

Jmenné prostory

Jmenné prostory jsou inovativní cestou jak kontrolovat přehlednost a viditelnost deklarací. Jsou podobné klasickým specifikům přístupu použitých ke kontrole viditelnosti jako třeba `public`, `private` nebo `protected`, nicméně jsou uživatelsky volitelné, takže mohou být pojmenovány jakkoliv.

Flex například používá jmenný prostor `fx_internal` pro svoje interní data. Systém jmenných prostorů je vybaven univerzálním identifikátorem zdrojů, což je prostředek, který kontroluje, aby nedošlo ke kolizím mezi zdroji. Jmenné prostory jsou také používané při práci s XML pomocí E4X. Systémem jmenných prostorů je vybavena většina moderních objektově orientovaných jazyků jako je například Java nebo C#, proto bylo logické, že vývojáři pro jazyk ActionScript 3.0 tento systém také zvolili.

Primitivní datové typy

ActionScript 2.0 měl jediný numerický typ a to Number, což bylo reálné číslo s dvojitou přesností. Pro řádné programování však toto již nestačilo, a proto byl přidány dva celočíselné typy int a uint. Oba dva jsou 32 bitové a umožňují procesoru mnohem rychlejší výpočty, když počítá bez desetinné čárky. Celočíselný typ je také mnohem vhodnější pro pomocné výpočty v algoritmech, jako jsou například počítadla v cyklech. Uint se od typu druhého liší pouze tím, že jeho rozsah nevstupuje do záporných čísel, ale začíná na nule. Díky tomu má ovšem dvojnásobný rozsah ve směru kladných čísel.

4.3.3. Flash Player API

Flash Player API je sada tříd a funkcí, které ActionScriptu dávají k dispozici třídy a funkce přehrávače Flash Player. Tato funkcionalita je mostem mezi jádrem jazyka ActionScript a zbytkem platformy Flash. Je zdrojem velké síly dostupné aplikacím stojícím na Flashi a je to velice důležitý dodatek k jádru jazyka ActionScript. I když tady není místo k popisu úplně každé nové funkce dostupné vývojářům, proberu zde alespoň ty nejvýznamnější z nich.

DOM3 model událostí

Model událostí Document Object Model Level 3 neboli DOM3 poskytuje standardní způsob generování a obsluhy událostí a jejich zpráv. Díky tomu mohou objekty ve Flash aplikacích mezi sebou komunikovat, navzájem spolupracovat, udržovat si informace o svém stavu a reagovat na změny. Tento model poskytuje průhlednější a efektivnější mechanismus správy událostí, než byl v předchozích verzích ActionScriptu.

Události a chyby jsou lokalizovány v balíčku flash.events.package. Rozhraní obsluhující všechny Flash komponenty používá úplně stejný model událostí jako má Flash API, a proto je systém událostí stejný v celé platformě Flash.

API pro objekty na ploše

Jedná se o API, která obsluhuje přístup k seznamu zobrazených objektů ve Flash Playeru. Jde o stromovou strukturu, ve které jsou uloženy všechny viditelné objekty

aplikace. Celá API se skládá z nových funkcí, které pracují s jednoduchými grafickými komponentami ve Flashi.

Například nová třída `Sprite`, o které jsem se již v této práci zmiňoval, je odlehčenou verzí třídy `MovieClip`. Je jí velice podobná ale více se hodí ke konstrukci grafických uživatelských rozhraní aplikace. Nová třída `Shape` pak reprezentuje základní vektorové tvary. Tyto třídy mohou být dynamicky vytvořeny pomocí operátoru `new` a mohou jim být také kdykoliv dynamicky změněny rodiče. Další novinkou je nová správa měření „hloubky“ ve Flash Playeru. To obsahuje nové metody pro práci se zetovou souřadnicí objektů a jejich pořadí ve vrstvách.

Obsluha dynamických dat a obsahu

ActionScript 3.0 obsahuje mechanismy pro načítání a obsluhu zdrojů a dat ve vaší Flash aplikaci tak, že tyto mechanismy jsou intuitivní a jsou konzistentní napříč celým Flash Player API, ať už se jedná o jakýkoliv typ dat.

Nová třída `Loader` poskytuje jednoduchý mechanismus pro načítání SWF souborů a obrázků a poskytuje přístup k informacím o načteném obsahu. Třída `URLLoader` poskytuje samostatný mechanismus pro načítání textu a binárních dat pro aplikace, které se zaměřují na jejich zpracování. Třída `Socket` pak poskytuje programátorovi způsob načítání a ukládání binárních dat na server v jakémkoliv formátu.

Nízkoúrovňový přístup k datům

Několik různých API poskytuje nízkourovňový přístup k datům tak, jako to doposud v ActionScriptu nebylo možné. Například pro data, která se právě stahují, poskytuje třída `URLStream`, která je implementovaná v již zmíněné třídě `URLLoader`, přístup přímo k již staženým binárním datům. Třída `ByteArray` pak nechá programátora optimalizovat čtení, zápis a práci s binárními daty. Nové zvukové API pak poskytuje detailní kontrolu zvuku pomocí tříd `SoundChannel` a `SoundMixer`. A v neposlední řadě nové bezpečnostní API, které poskytuje bezpečnostní informace a uživatelská práva SWF souboru a načteného obsahu a umožňuje programátorovi lépe zvládat bezpečnostní chyby

Práce s textem

ActionScript 3.0 obsahuje balíček `flash.text`, který využívají všechny API, jenž mají jen vzdáleně něco společného s textem. Třída `TextLineMetrics` poskytuje detailní informace o řádce textu v textovém poli a nahrazuje `TextField.getLineMetrics()` metodu známou z ActionScriptu 2.0 . Třída `TextField` nyní obsahuje celou řádku zajímavých funkcí, které slouží k poskytování specifických informací o řádce textu anebo jednom znaku v textovém poli.

Příkladem těchto metod je například `getCharBoundaries()`, která vrátí obdélník ohraničující znak. Dále pak `getCharIndexAtPoint()`, vracející index znaku na určitém bodě obrazu a pak `getFirstCharacterInParagraph()`, která vrátí index prvního znaku v odstavci. Z těch, které pracují na úrovni řádku, mohu pak vyjmenovat například klasickou `getLineLength()`, vracející délku řádku a `getLineText()`, která vrátí celý text specifikovaného řádku v odstavci. A naposled tu zmíním třídu `Font`, která se stará o správu fontů defaultně vložených do SWF souborů

5. Využití ActionScriptu 3.0

Tato část se nejvíce z celé práce bude věnovat praktické aplikaci. Práce není zamýšlena jako učebnice programování ani manuál jazyka, má spíše nastínit možnosti ActionScriptu 3.0 a odůvodnit, proč by si ho měl programátor vybrat v dnešní konkurenci prostředků pro tvorbu animací a bohatých internetových aplikací.

Celá tato část bude v podstatě sestávat z praktických ukázek nastiňujících možnosti jazyka názorně namísto suchého popisu. Všechny ukázky budou pak jako zdrojový soubor ActionScriptu a soubor SWF uloženy na CD jako příloha k práci, aby si případný zájemce mohl projít kód a zjistit, jak konkrétně bylo kýženého efektu docíleno. Pokud se v adresáři nachází HTML soubor spouští se ukázka pomocí něj.

Všechny ukázky jsou pak stvořeny podle manuálů k jazyku ActionScript z různých učebnic a stránek, zabývajících se touto tematikou. Tyto stránky jsem pak uvedl ve zdrojích a někdy jsou obsaženy přímo v samotné animaci. Při výběru konkrétních ukázek jsem se pak řídil jednak názorností ukázky a toho, jak moc demonstruje sílu jazyka, tak samotným efektem, který demonstruje. Snažil jsem se vybírat takové efekty, které jsou dnes uživateli internetu žádané a oblasti, ve kterých ActionScript 3.0 obzvláště vyniká.

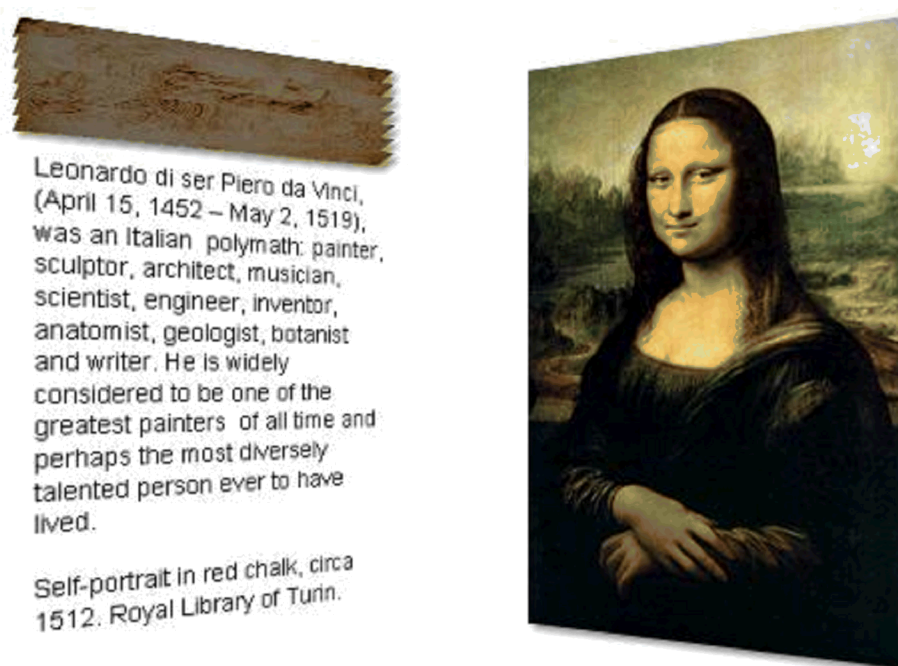
5.1. Pohybové efekty

Jedná se hlavně o efekty pomocí vylepšeného systému Motion Tween o kterém jsem se již zmiňoval. Tyto efekty se používají především ve hrách anebo v aplikacích pro přidání interaktivity či dramatického efektu. Poslední dobou se také na internetu často objevují velice pohyblivé a interaktivní reklamní bannery ve formě her.

5.1.1. Skládací obrazy

Effektu složení obrazu anebo jiného jeho pohybu lze docílit pomocí třídy ImageAccordition. V ukázce dochází ke složení a rozložení obrazu podle předem zadaných kritérií. Využívá se u toho také 3D API, kterou obsahuje Flash Player verze 10. V ukázce stačí kliknout na složený anebo rozložený obraz myší a ten se pak plynulým pohybem změní. Tohoto efektu lze například využít, pokud chceme ušetřit místo na ploše aplikace a některé objekty tak zakryjeme jinými. Náhled z ukázky je

vidět na obrázku 4. Celá ukázka je dostupná na přiloženém CD v adresáři SkladaciObrazy.



Obrázek 4 Efekt skládání obrazu

5.1.2. Gumový obraz

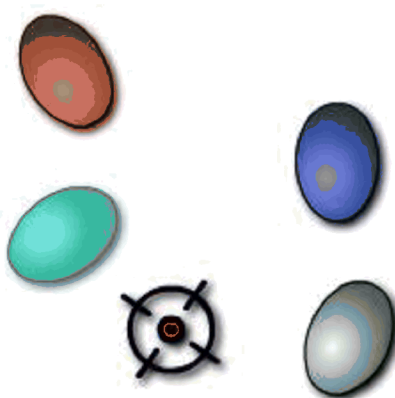
Tohoto efektu je dosaženo pomocí třídy BitmapTransformer, která je součástí třídy Tween. Nejprve je použito transformeru tak, aby bylo možné obraz libovolně natáhnout chycením za jeden z jeho rohů a poté je použita třída Tween k tomu, aby opětovným propuštěním bodu, za který obraz natahujeme, se obraz vrátil „gumovým“ efektem do původního stavu. Tyto efekty transformace obrazu, i když ne tak dramatické, se používají především tam, kde je v aplikacích možnost si uspořádat komponenty na ploše. Náhled z ukázky je vidět na obrázku 5 a celá ukázka je dostupná v adresáři GumovyObraz.



Obrázek 5 Efekt gumového obrazu

5.1.3. Rotace objektů mimo jejich střed

Jedna z nových zajímavých možností ActionScriptu 3.0 je rotace objektu tak, aby byl jako střed rotace brán v potaz jiný bod než střed otáčeného tělesa. Pokud chceme využít možnosti rotace jakéhokoli tělesa na scéně, můžeme využít jeho atribut `rotation`. Při jeho změně se pak těleso otočí podle zadaného úhlu. Každá rotace musí mít svůj střed a při otáčení pomocí ActionScript se za střed považuje střed otáčeného tělesa. Pokud ovšem chceme otáčet kolem jiného bodu, musíme k tomu využít třídu `MatrixTransformer` a její metodu `rotateAroundExternalPoint`. V ukázce je možné pohybovat myší všemi tělesy včetně středu otáčení a otáčet pak označeným tělesem pomocí šipek vlevo a vpravo. Na obrázku 6 je pak vidět náhled této ukázky a celou ji opět nalezneme na přílohovém CD v adresáři `RotacniPohyb`.



Obrázek 6 Rotace objektů mimo střed

5.1.4. Carousel menu

S nástupem Windows 7 se stalo jak na internetu, tak v různých aplikacích populární takzvané Carousel otočené menu. V následující ukázce je vidět, jak je možné bez spolupráce s jakýmkoliv 3D enginem donutit ActionScript 3.0 ke stvoření takového efektu. V podstatě se jedná o rozpohybování jednotlivých objektů MovieClip, do kterých předtím načteme jednotlivé obrázky. Více je vidět ve zdrojovém kódu ukázky, která je uložena v adresáři Carousel na přiloženém CD. Náhled je pak vidět na obrázku 7.



Obrázek 7 Carousel menu

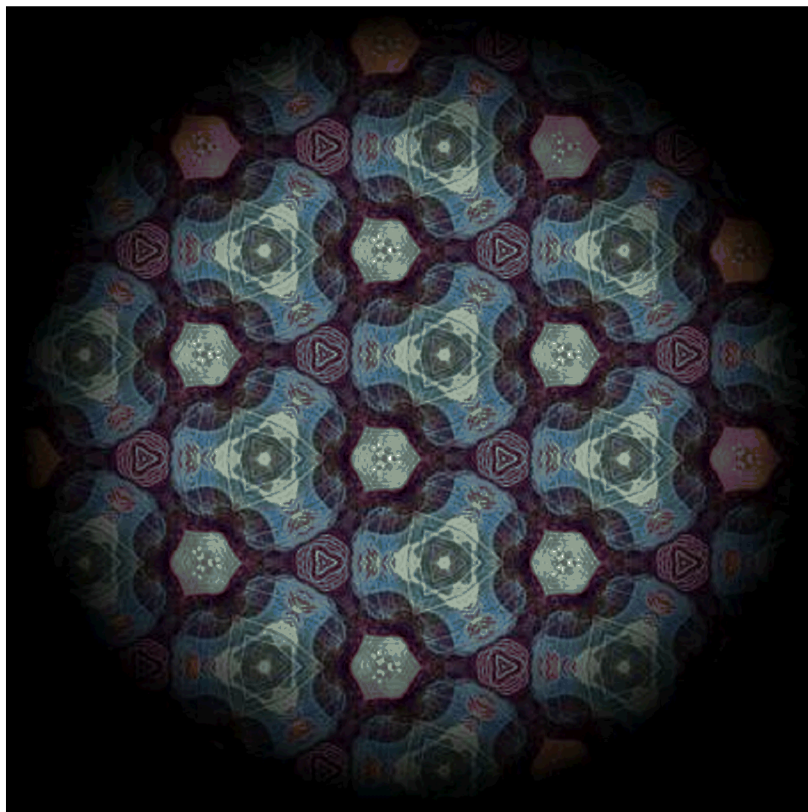
5.2. Grafické efekty

V ActionScriptu 3.0 je mnoho nových funkcí pro práci s grafikou. Vektorová grafika se velmi hodí pro různé úpravy a pohyby obrazů založené na matematických výpočtech. Dobrým příkladem toho je generování kaleidoskopických obrazů tak, jak tomu je v první ukázce této kapitoly. Druhá pak ukáže dnes například v reklamních bannerech velmi používaný efekt lupy, který pracuje s rastrovým obrazem.

5.2.1. Kaleidoskopické obrazy

Tato ukázka stojí na třídě KaleidoscopeThreeMirrors, která generuje pomocí zrcadlového efektu obraz klasického kaleidoskopu, na který jsme zvyklí z dětských hraček. Matematika celé věci stojí na rozložení obrazu do věnců stvořených

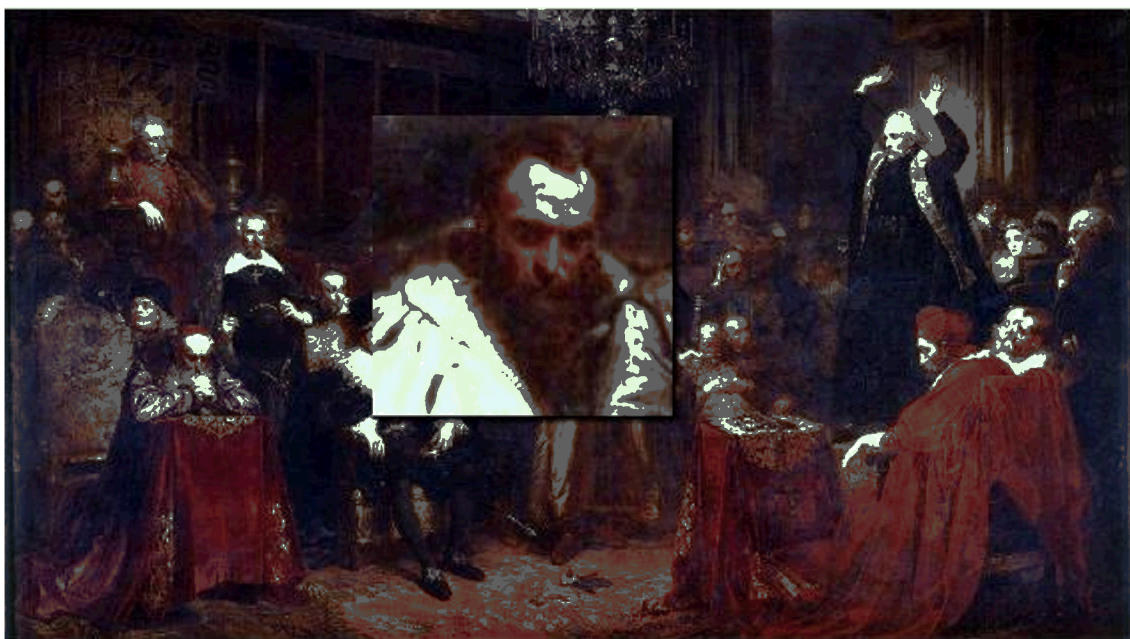
z trojúhelníků, kde dva sousední trojúhelníky vždy zrcadlí odrazy svých sousedů. V ukázce je pak možnost zvolit, zda chceme kaleidoskop tvořit na základě vektorového obrázku anebo rastrového. Náhled využívající rastrový efekt je pak vidět na obrázku 8. Celý projekt se pak nachází v příloze v adresáři Kaleidoskop.



Obrázek 8 Kaleidoskopický obraz

5.2.2. Efekt lupy

Tato ukázka stojí na základních funkcích ActionScriptu, které umožňují filtrování obrazu a jeho následnou úpravu. Při stisku tlačítka dochází k zobrazení čtverce, ve kterém se nachází zvětšený obraz pozadí. Tento efekt je dobře známý z několika vektorových grafických editorů a používá se docela často v internetových aplikacích tam, kde je nutné zaměřit uživatelovu pozornost na určitý detail, případně nechat ho volně prozkoumat objekt jeho zájmu. Jedná se o velice jednoduchou aplikaci, která může při drobné úpravě načítat a zvětšovat podle několika kritérií. Celý projekt ukázky včetně zdrojových souborů můžete nalézt v příloze v adresáři Lupa. Náhled je pak vidět na obrázku 9.



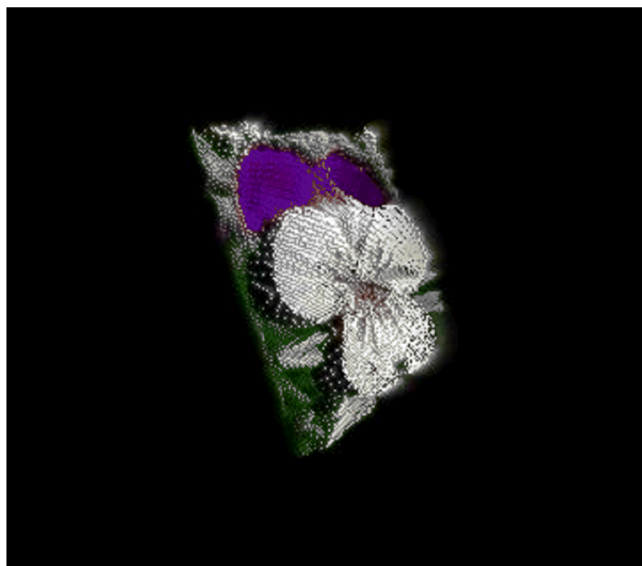
Obrázek 9 Efekt lupy

5.3. Částicové efekty

Tento druh efektů sestává většinou z rozložení obrazu na stovky až tisíce částic. Částice mohou být založeny buď na vektorech anebo na pixelech. Tento druh efektů se většinou masově nepoužívá, ale je velice efektivní a jedna z oblastí využití jsou pro něj například simulační Applety, které popisují fyzikální a chemické jevy. I když existují například aplikace, které tyto efekty používají jako efektivní přechod mezi obrazy, kdy se jeden rozpadne na malé kousky a pod ním se objeví druhý.

5.3.1. Rozklad obrazu

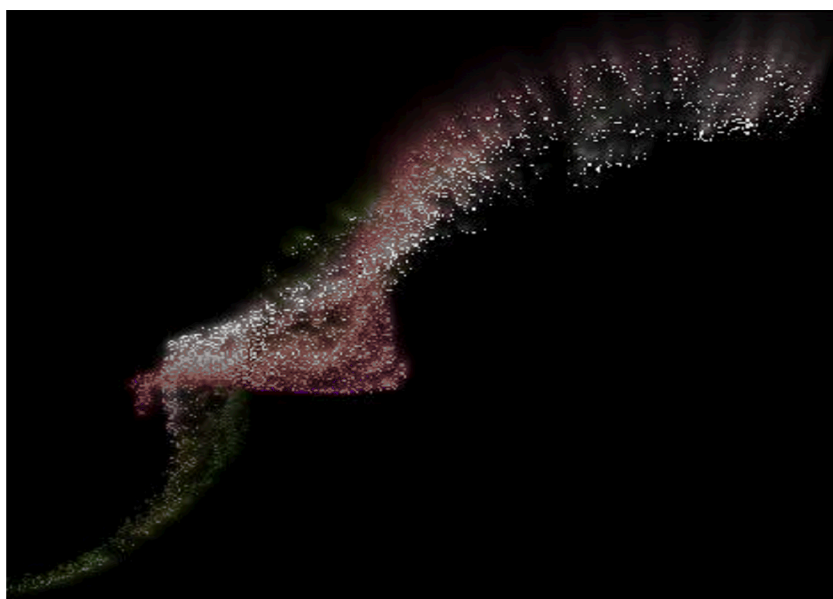
Tento efekt využívá rozkladu obrazu na bázi pixelů. V podstatě jde o obrázek, který se pohybuje v prostoru a jednou za čas se rozpadne na stovky částíček, které se podle určitého algoritmu zase po čase složí zpátky v původní obraz. Tato ukázka je velice efektivní, i když docela nepraktická a dovedu si představit její použití jen jako šetřič obrazovky. Nicméně efektivně demonstruje schopnost ActionScriptu pouze pomocí kódu předepsat matematicky rozklad obrazu a následný pohyb částic bez většího nároku na procesor. Celou ukázkou můžete najít v příloze v adresáři RozkladObrazu a náhled je pak vidět na obrázku 10.



Obrázek 10 Rozklad obrazu

5.3.2. Pohyb částic

Tato aplikace je v podstatě rozšířenou verzí předešlé ukázky. Na rozdíl od ní však zde je umožněno uživateli kontrolovat pohyb částic v několika různých směrech a tvořit tak seskupení částic libovolných tvarů. Na této ukázce je dobře demonstrováno, jak ActionScript umožňuje vytvářet simulace pohybu částic podle určitého matematického modelu. Celá, velice interaktivní, aplikace je včetně zdrojového projektu k dispozici v příloze v adresáři PohybCastic. Náhled na jeden z tvarů je pak vidět na obrázku 11.



Obrázek 11 Pohyb částic

6. Závěr

V této práci jsem si dal za úkol prozkoumat platformu Flash a především její jazyk ActionScript v současné verzi 3.0. Úkol to byl nelehký, protože se ukázalo, že jazyk sám o sobě je sice důležitý, ale že je třeba brát zřetel na spoustu jiných okolností. Práce jako taková byla směřovaná na ty, kdo se zajímají se o jazyk ActionScript a hlavně přemýšlející o tom, co nového přinesla takzvaná revolučnost jeho třetí verze.

Myslím, že tyto cíle jsem splnil a že tato práce obsahuje spoustu informací o jazyce ActionScript a prozkoumává ho z několika hledisek. Ať už to je celá platforma Flash, která na jazyce stojí, anebo jeho vývoj počínaje první verzí jazyka. Podrobně jsem popsal také Flash Player, ten „zázračný“ přehrávač, bez kterého už si dnešní uživatelé internetu svůj život ani nedokáží představit.

Několikrát jsem také předestřel užitečnost a schopnosti jazyka jak slovním popisem, tak konkrétními ukázkami oblastí jeho využití. Těchto oblastí je skutečně mnoho a pokud se mám ještě naposledy zamyslet, myslím, že i například ve školství je plno oblastí, kde by tento jazyk a platforma Flash našly uplatnění. Ať už to jsou e-learningové kurzy anebo animace popisující nějaký fyzikální děj, ActionScript je vždy schopen se prosadit na poli multimediální animace.

Rozdíl mezi druhou a třetí verzí jazyka je markantní a pro čtenáře, které zajímalo konkrétně toto, jsem připravil nespočet srovnání a odborných náhledů na problém srovnání těchto dvou verzí. Ačkoliv programátor seznámený s verzí 2.0 bude mít trochu problém s přestupem na verzi novější, začátečník úplně neznalý problematiky, byť zkušený v oblasti programování, nalezne ve verzi 3.0 velmi silný nástroj, který mu po krátkém zaučení přinese možnosti, o kterých se mu na poli interaktivní animace a bohatých internetových aplikací ani nesnilo.

7. Seznam použitých zdrojů

1. **Pwso.** Adobe Flash CS3 Professional [Online] Pwso blog, 17. leden 2008. [Citace: 20. červen 2010.] <http://pwso.blogspot.com/2008/01/adobe-flash-cs3-professional.html>
2. **KEN.** Adobe Flex Builder Professional [Online] FreshWap [Citace: 21. červen 2010.] <http://www.freshwap.net/applications/607-adobe-flex-builder-professional-v3.0.2-incl.html>
3. Adobe AIR [Online] Qweas, 26. únor 2010. [Citace: 19. červen 2010.] <http://www.freshwap.net/applications/607-adobe-flex-builder-professional-v3.0.2-incl.html>
4. Adobe Flash File Extensions [Online] File Extensions [Citace: 10. červen 2010.] <http://www.file-extensions.org/adobe-flash-file-extensions>
5. **Steve Jobs.** Flash odmítáme z technických důvodů, ne z obchodních [Online] ITBiz, 30. duben 2010. [Citace: 15. červen 2010.] <http://www.itbiz.cz/steve-jobs-flash-odmitame-z-technicky-duvodu-ne-z-obchodnich>
6. Programming ActionScript 3.0 [Online] Adobe [Citace: 10. červen 2010.] http://livedocs.adobe.com/flex/3/html/help.html?content=Part6_ProgAS_1.html
7. **Scott Manning.** 100% Pure Flash - What Can Flash Achieve Alone? [Online] SitePoint, 29. květen 2003. [Citace: 10. červen 2010.] <http://articles.sitepoint.com/article/can-flash-achieve-alone>
8. **Bambi Turner.** The History of Macromedia Flash [Online] eHow [Citace: 5. červen 2010.] http://www.ehow.com/about_5257148_history-macromedia-flash.html
9. Versions of ActionScript [Online] ActionScriptVersions [Citace: 18. červen 2010.] <http://www.actionscriptversions.com/>
10. **Calif.** Adobe Completes acquisition of Macromedia [Online] Adobet, 5. prosinec 2005. [Citace: 10. červen 2010.] <http://www.adobe.com/aboutadobe/pressroom/pressreleases/200512/120505AdobeAcquiresMacromedia.html>
11. **Garry Grossman, Emmy Huang.** ActionScript 3.0 Overview [Online] Adobe, 26. červen 2006 [Citace: 20. červen 2010.] http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html
12. Flash Player Version History [Online] MediaCollege [Citace: 12. červen 2010.] <http://www.mediacollege.com/adobe/flash/player/version/>

13. Flex 3 Function Parameters [Online] Adobe [Citace: 16. červen 2010.] http://livedocs.adobe.com/flex/3/html/help.html?content=03_Language_and_Syntax_19.html
14. **Lee Brimelow**. Six reasons to use ActionScript 3.0 [Online] Adobe, 18. červen 2008 [Citace: 15. červen 2010.] http://www.adobe.com/devnet/actionscript/articles/six_reasons_as3.html
15. Standard ECMA-262 [Online] Ecma International [Citace: 18. červen 2010.] <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
16. Adobe acknowledges critical security flaw in software [Online] BBC News, 7. červen 2010 [Citace: 16. červen 2010.] <http://news.bbc.co.uk/2/hi/technology/10257411.stm>
17. How to manage and disable Local Shared Objects [Online] Adobe [Citace: 18. červen 2010.] <http://kb2.adobe.com/cps/526/52697ee8.html>
18. Flash Platform [Online] FlashPlatform [Citace: 5. červen 2010.] <http://www.flashplatform.cz/>
19. **Chris Kaminski**. Adobe, 'Rich Internet Applications' and Standards [Online] Webstandards, 3. duben 2005 [Citace: 15. červen 2010.] <http://www.webstandards.org/2005/04/19/adobe-rich-internet-applications-and-standards/>
20. **Dan Gries, Barbara Kaskosz**. 3D Accordion Image Effect in AS3 [Online] FlashAndMath [Citace: 23. červen 2010.] <http://www.flashandmath.com/flashcs4/leo/index.html>
21. **Barbara Kaskosz**. Gummy Bitmap [Online] FlashAndMath [Citace: 21. červen 2010.] <http://www.flashandmath.com/intermediate/gummy/example2.html>
22. **Ondřej Brichta**. Rotace objektů mimo jejich střed [Online] FlashCz 23. březen 2009. [Citace: 21. červen 2010.] <http://www.flash.cz/portal/clanek.aspx?id=1390>
23. **Ondřej Brichta**. Carousel menu pro AS3 [Online] FlashCz 26. říjen 2009. [Citace: 21. červen 2010.] <http://www.flash.cz/portal/clanek.aspx?id=1556>
24. **Dan Gries, Barbara Kaskosz**. The Kaleidoscope [Online] FlashAndMath [Citace: 23. červen 2010.] <http://www.flashandmath.com/flashcs4/kal3mir/index.html>
25. **Barbara Kaskosz**. Magnifying Glass Effect [Online] FlashAndMath [Citace: 27. červen 2010.] <http://www.flashandmath.com/intermediate/magglass/index.html>
26. **Dan Gries**. The Image Exploding into Particles [Online] FlashAndMath [Citace: 26. červen 2010.] <http://www.flashandmath.com/advanced/particles3d/>

27. **Dan Gries.** Flower Twist Particle Effect [Online] FlashAndMath [Citace: 25. červen 2010.]
<http://www.flashandmath.com/flashcs4/flower/index.html>

8. Seznam obrázků

Obrázek 1 Vývojové prostředí Adobe Flash CS3, zdroj: (1)	14
Obrázek 2 Flex Builder verze 3, zdroj: (2)	16
Obrázek 3 Ukázka aplikací Adobe Integrated Runtime, zdroj: (3).....	17
Obrázek 4 Efekt skládání obrazu	43
Obrázek 5 Efekt gumového obrazu	44
Obrázek 6 Rotace objektů mimo střed	44
Obrázek 7 Carousel menu	45
Obrázek 8 Kaleidoskopický obraz	46
Obrázek 9 Efekt lupy.....	47
Obrázek 10 Rozklad obrazu.....	48
Obrázek 11 Pohyb částic.....	48

9. Seznam tabulek

Tabulka 1 Typy souborů platformy Flash, zdroj: (4)	7
---	----------

10. Seznam příloh

Příloha A: Praktické ukázky aplikací ve Flashi, jejich zdrojové kódy a projekty (CD-ROM)

11. Cizojazyčné resumé

This work is focused mainly on ActionScript 3.0 and its implications to the whole Flash platform. It provides many valuable information about this newest version of the object oriented programming language. The ActionScript and the Flash Player are the cornerstones of the entire Flash platform and this work justly gives them enough room.

In the first part there is much room presented for the entire Flash platform and its connection to ActionScript. The chapter focuses on the fundamentals about flash platform. For example it provides much information on the Flash Player, authoring tools and usability of the entire platform.

The second part focuses mainly on history and follows the development of the entire platform with the focus on ActionScript and its earlier versions. At the end of the chapter there is much information about previous versions of ActionScript and their comparison.

The third chapter explore the entire programing language version 3.0 from the programmers kind of view. It reveals much information about innovations in the language. The chapter is divided by angles of look at the programming language and its new capabilities.

The last part is focused on practical implications of the language and it provides many examples of its usability. The concrete examples are available to the reader on the Cd-Rom inserted to this work.