

Multimediální učebnice technologie obrábění

Multimedial encyclopedia of machining technology

Bc. Robert ŠOMAN

Diplomová práce
2007



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2006/2007

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Robert ŠOMAN**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Multimediální učebnice technologie obrábění**

Zásady pro vypracování:

1. Vypracujte přehled technologií e-learningu a multimediální výuky
2. Zvolte vhodnou technologii pro tvorbu multimediální učebnice
3. Vytvořte multimediální učebnici obrábění
4. Obsah učebnice bude pokrývat látku jednoho školního roku
5. Učebnice bude obsahovat animace a videosekvence postupů při práci

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ULRICH, K. **Macromedia Flash MX 2004** Názorný průvodce. Computer Press: 2005.
2. FRANKLIN, D., MAKAR, J. **Flash MX 2004 Actionscript** – oficiální výukový kurz. SoftPress: 2005.
3. DeHAAN, J. **Macromedia Flash MX 2004**, oficiální výukový kurz. SoftPress: 2004.

Vedoucí diplomové práce:

Ing. Pavel Vařacha

Ústav aplikované informatiky

Datum zadání diplomové práce:

13. února 2007

Termín odevzdání diplomové práce:

28. května 2007

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Ing. Ivan Zelinka, Ph.D.

ředitel ústavu

Abstrakt

Důvodem této práce byla absence učební pomůcky pro výuku Technologie obrábění. Po důkladném zvážení všech pro a proti se autor (sám učitel odborných předmětů na střední škole) rozhodl vytvořit si takovou pomůcku v elektronické podobě sám. Vytvořil tedy pomocí programu FLASH „Multimediální technologii obrábění pro 3. ročník oboru Mechanik – seřizovač“. V první části se zabývá obecně elektronickými učebními pomůckami (tzv. E-learningem) – jednotlivými typy, jejich, výhodami a nevýhodami. Po rozhodnutí o nejjednodušší technologii (program FLASH) se pak věnuje obecnému popisu programu. Praktická část je zaměřena na popis klíčových postupů při tvorbě samotné prezentace.

Klíčová slova

Flash, multimedia, e-learning, výuka, encyklopedie

Abstract

Reason for this work was absence of instructional setout for education technology cutting. After thorough weigh of all pro and con with author (he is alone a teacher of this speciality on secondary school) decided to build up such setout in electronic shape alone. Fecit then by the help of programme FLASH „Multimedia technology cutting for 3. vintage enclosure mechanic – setter“. In the first part it deals with generally electronic instructional help (so-called E-learningem) – single print, their advantage and disadvantage. After determination about optimal technology (programme FLASH) with then inscribe to common program description. Practical part is sight on description pivotal progress at production presence himself.

Keywords

Flash, multimedia, e-learning, learning, encyclopedia

Na tomto místě bych chtěl poděkovat Ing. P. Vařachovi (vedoucímu mé diplomové práce), za velkou pomoc a konstruktivní připomínky při tvorbě této práce a také své rodině za psychickou podporu a trpělivost s jakou snášely mé časté večery u počítače.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....

Podpis diplomanta

OBSAH

ÚVOD.....	6
E-LEARNING A MULTIMEDIÁLNÍ VÝCHOVA.....	9
1 POPIS EXISTUJÍCÍCH ŘEŠENÍ.....	11
1.1 SW řešení.....	11
1.2 Hardware.....	13
2 VÝBĚR OPTIMÁLNÍHO ŘEŠENÍ.....	17
2.1 Software.....	18
2.2 Hardware.....	19
2.3 Shrnutí.....	19
FLASH (PŘEDSTAVENÍ PROGRAMU).....	21
1 FILOZOFIE PRODUKTU.....	21
2 PRACOVNÍ PROSTŘEDÍ.....	22
2.1 Paletky nástrojů.....	23
2.2 Časová osa (Timeline).....	25
2.3 Scéna.....	26
2.4 Vlastnosti (Properties).....	27
2.5 Nástroje (Tools).....	27
3 KRESLENÍ.....	29
3.1 Čára (Line).....	32
3.2 Obdélník (Rectangle).....	32
3.3 Polygon (Polygon).....	35
3.4 Ovál (Oval).....	35
4 TEXTY.....	35
5 SYMBOLY.....	41
5.1 Klip (Movie Clip).....	44
5.2 Tlačítko (Button).....	44
5.3 Grafika (Graphic).....	46
5.4 Úpravy instancí symbolů, Efekty.....	46
6 KNIHOVNY.....	48
6.1 Význam, funkce.....	48
6.2 Sdílení knihoven.....	49
6.3 Import.....	50
7 ANIMACE.....	52
7.1 Animace políčko po políčku.....	52

7.2	Animace pomocí Tween shape.....	52
7.3	Animace pomocí Motion tween.....	53
8	PUBLISHING.....	54
8.1	Export do .swf.....	55
8.2	Export do .html.....	55
9	ACTION SCRIPT.....	56
	TVORBA ENCYKLOPEDIE.....	59
1	NÁVRH ŘEŠENÍ.....	59
2	Hlavní program.....	62
2.1	Hlavička.....	63
2.2	Záložky.....	63
2.3	Tlačítka.....	65
2.4	Časové osa.....	66
2.5	Loader.....	68
3	PREZENTACE OBRAZOVEK.....	70
3.1	Rodičovská obrazovka.....	70
3.2	Dceřiné obrazovky.....	73
4	TEXTOVÁ ČÁST.....	75
4.1	Text – informační obsah.....	75
4.2	Text – navigace.....	76
5	OBRÁZKY.....	76
5.1	Obrázky z Flashe.....	76
5.2	Obrázky importované.....	78
6	ANIMACE.....	79
6.1	Tangenciální nůž.....	79
7	VIDEOSEKVENCE.....	81
8	AUDIO.....	81
9	WEBOVÉ SLUŽBY, FORMULÁŘE.....	82
10	PUBLIKOVÁNÍ.....	85
	SEZNAM POUŽITÉ LITERATURY.....	88
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	90
	SEZNAM OBRÁZKŮ.....	91
	SEZNAM TABULEK.....	95

ÚVOD

E-learning (nebo také eLearning) je dnes slovo skloňované ve všech pádech. Každý chce mít e-learningový kurz a být „in“. A hodně lidí tuší (a vcelku oprávněně), že se zde nachází pole neorané pro obrovské zisky.

Bohužel všechny tyto aktivity jsou směřovány na nejrozšířenější oblasti výuky (především na výuku cizích jazyků a výpočetní techniky – tedy ty, které jsou v kurzu) a do jiných odvětví se dostává velmi těžko.

E-learning je relativně levný, ovšem pouze tehdy, když se do něj zapojí větší množství klientů. Ve školství (kde pracuji i já) je však vyučující silně znevýhodněn:

- Počet „klientů“ je omezen (ročně nepřesáhne řádově 100 studentů).
- E-learning odborných předmětů je drahý, protože se obvykle vyrábí na zakázku a škola na nákup drahých programů peníze nemá.

Po tomto zjištění zůstávají kantorovi pouze tři řešení:

1. Smířit se se stávající situací a používat dále „klasické“ způsoby (bohužel tak přijde o mnoho výhod plynoucích z použití e-learningu).
2. Sehnat peníze neustálým obtěžováním vedení školy (a stát se tak „velmi oblíbeným“)
3. Rozhodnout se, že si pomůcky vytvoří sám (to je také můj případ).

Nad tvorbou podobného programu jsem uvažoval již několik let, ale chyběl mi nějaký počáteční impulz. Tím se stala tato DP při mém studiu informatiky na UTB ve Zlíně.

Tato práce je tedy zaměřena na e-learning, speciálně na LCMS (tvorba materiálů pro e-learning).

Nejprve se obecně zmiňuje o problematice (1. kapitola Teoretické části), kde také představuje různé možnosti elektronické výuky dnes využívané ve školství. V druhé části se již věnuje produktu firmy Adobe – Flashi, který jsem si pro tvorbu učební pomůcky vybral. Důvody tohoto výběru uvádím v závěru 1. kapitoly.

Celá druhá kapitola je zaměřena na popis technik tvorby použitých při práci.

Hlavním cílem této práce byla především tvorba jakési encyklopedie pro potřebu nejen mou, ale i dalších kantorů, kteří řeší stejný problém.

I. TEORETICKÁ ČÁST

E-LEARNING A MULTIMEDIÁLNÍ VÝCHOVA

E-learningem rozumíme v podstatě jakoukoliv výuku, při které použijeme jakékoliv elektronické pomůcky [1]:

„... eLearning je v podstatě jakékoli využívání elektronických materiálních a didaktických prostředků k efektivnímu dosažení vzdělávacího cíle s tím, že je realizován zejména/nejenom/ prostřednictvím počítačových sítí ... „

další definici nabízí např. [2] :

Řečeno stručně a velmi jednoduše, e-Learning není nic jiného než efektivní využívání informačních technologií v procesu vzdělávání.

E-learning se pak dále dělí [3] a [4]:

- Blended Learning – obvykle se používá společně s dalšími typy výuky (typicky je to učitel). Cílem je integrace výuky do každodenního života. Tato kategorie je náročná na citlivé rozložení výuky mezi jednotlivé typy. Blended learning se dále dělí na
 - Synchronní výuku – probíhá v reálném čase za fyzické účasti více studentů, takže si navzájem vyměňují poznatky – klasická výuka v učebně
 - Asynchronní výuku – je zaměřena na jednotlivé studenty, kteří tak mohou pracovat vlastním tempem a v čase, který jim lépe vyhovuje – typicky se jedná o různá výuková CD, audio, video ...
- Klasická výuka – přestože je tato výuka zaměřena na elektronické materiály, bez lektora se neobejde. Ten pak může využívat možnosti elektronické komunikace – videokonference, email, hlasové komunikace apod. Obrovský potenciál také skýtá tzv. „elektronické přeškolení“ - studentům jsou předem distribuovány elektron. materiály, takže při setkání s lektorem již studenti ovládají alespoň základy problému.
- Clear eLearning – veškerá výuka je založena na elektronické komunikaci (od přihlášení až po závěrečné testy) a student se s lektorem vůbec nemusí setkat.

Dále můžeme na [5] zjistit, že e-learningové systémy se dělí na dvě základní skupiny:

- LMS – systém řízení výuky
- CLMS – elektronické systémy pro tvorbu kurzů

Přínosem e-learningu by mělo být zvýšení komunikace studentů, usnadnění a urychlení přístupu k informacím, v neposlední řadě také vedení k týmové práci. Možností interaktivity a pocitu, že záleží i na nich (studentech) se několikanásobně zvyšuje efektivita celého procesu .

Proti klasickému vzdělávání má e-learning mnoho dalších předností – podklady stačí vytvořit jen jednou, modifikace a modernizace je jednoduchá, odpadají náklady na další lektory, pronájem prostor apod. Pro firmy je také podstatné, že při e-learningovém školení nemusí být zaměstnanci vyřazeni z pracovního procesu (snížení skrytých ztrát). Školení není vázáno na dostatečný počet studentů, odezva na změny je rychlá. Studenti mohou pracovat svým tempem, látku kterou již ovládají, mohou přeskočit (může být podmíněno

splněním testu z přeskokované látky), volit si pořadí lekcí apod.

V e-learningu existují podobně jako v dalších odvětvích i standardy AICC a SCORM.

AICC

AICC (Aviation Industry CBT Committee) [6] – dnes spíše historický model. Původně vznikl pro letecký průmysl, díky své kvalitě se rozšířil i do dalších odvětví. Přestože je již překonán, pro svou rozšířenost je při tvorbě e-learningových systémů stále vyžadován.

Obsah se dělí na :

- spustitelné jednotky (assignable units) – také nazývány jako lekce. Tyto jednotky jsou dále nedělitelné, skládají se z vlastního textu (obsahu) a integrovaného navigačního systému.
- kurzy (courses) – spustitelné jednotky sloučené do většího celku.

Kurzy lze dělit do tzv. bloků, které mohou být svázány ve více úrovních. Mezi lekcemi a bloky kurzu lze nastavovat logické vztahy pro podmínky pro přechod na další úroveň (např. splnění 60% při závěrečném testu apod.)

Existuje několik úrovní standardizace AICC, kvalitní systém by měl umět zpracovávat data o času práce, výsledky předešlých testů atp.

SCORM

SCORM (Sharable Content Object Reference Model) [7] – v současnosti nejpoužívanější systém. Obsah se skládá z učebních objektů (jednotka libovolné velikosti, která obsahuje výukovou informaci). Objektem může být jediná věta, obrázek, animace, ale i celý kurz.

Objekt se skládá z výukového obsahu a popisných dat (meta-data) pro specifikaci učebních oborů (účel, autor, popis, poznámky, ...). Navigaci zprostředkovává LMS SCORM kompatibilní s přehrávačem SCORM RTE.

SCORM musí mít konkrétní souborovou strukturu, především soubor *imsmanifest.xml* s popisem kurzu včetně vnitřní struktury.

Nakonec si dovoluji jednu kritickou poznámku: při dnešním trendu e-learningu se tento způsob protlačuje do výuky silou bez ohledu na vhodnost použití (alespoň ve školství – inu státní). Jedině ve škole jsme se setkali se silovým zaváděním e-learningu např. do češtiny, kde sice svůj význam má, ale nemůže nahradit klasickou výuku (mám na mysli základní a střední školy).

1 Popis existujících řešení

Pro popis existujících řešení je potřeba rozlišit dvě základní oblasti – HW a SW řešení.

1.1 SW řešení

Z hlediska certifikačních autorit (viz kap 1.) je SW řešení jediné možné. Jedná se tedy výhradně o programové vybavení, které nemusí být vázáno na konkrétní hardware. Jedná se především o:

Email

V počátcích výpočetní techniky to byla v podstatě jediná možnost e-learningu. V podstatě je logickým pokračováním původních korespondenčních kurzů. Změnilo se pouze komunikační medium. Přesto, že email byl velký skok, interaktivita byla minimální a komfortu dnešních systémů zdaleka nedosahuje. Příkladem může být např. [8].

Html+xhtml

Dnes velmi rozšířený způsob. Pomocí těchto protokolů lze distribuovat nejen text, ale připojit i video a audio soubory. Tento způsob lze využít k oslovení velkého množství zájemců, proti emailu je další marketingovou výhodou, že kurz přijde za vámi kdekoliv, kdykoliv. Vynikající ukázkou této technologie je na [9].

Databáze

Databáze sice neslouží primárně k výuce, ale využívají se spíše k ověření znalostí. V českém školství se mimo jiné využívá i program doSystem (dříve dotest) [10].

Programy pro e-learning

rozlišujeme 3 základní kategorie:

- obecné programy využitelné i pro tvorbu e-learningových materiálů – např. *PowerPoint* (balík MS Office), *Impress* (balík OpenOffice.org),... .

Tento způsob je učiteli hojně využíván pro dostupnost jednotlivých programů – MS Office jsou dnes standardním vybavením počítačů a OpenOffice.org jsou jeho freewarovou alternativou.

- programy pro tvorbu e-learningových materiálů – *SWISH, FLASH, CAPTIVATE, ...* . Tyto programy sice vyžadují hlubší znalosti prostředí, ovšem jejich grafické i programátorské jsou téměř neomezené. Odpadají jakékoliv problémy s přenosem na jiné platformy OS, budeme-li řešit programy s výstupem ve .flv nebo .swf (nativní formát Flashe, Swishe) nevznikají žádné nároky na další programové vybavení PC. Jediná podmínka je, aby na PC byl Flash player, který je ale standardní součástí každého OS, pokud není nainstalován, automaticky se nabízí ke stažení např. při návštěvě webových stránek s Flash obsahem atd. Také lze případnou kolizi vyřešit „přibalením“ přehrávače do distribuce materiálů, takže jsou dodávány ve formě .exe souborů.
- e-learningové programy – programy jsou určeny pro použití ve výuce. Jsou tedy již od začátku koncipovány pro určitý obor a oblast informací. Tyto produkty jsou obvykle zpracovány na vysoké úrovni, jejich cena může být ale velmi vysoká. Obměna nebo modernizace je omezená nebo přímo nemožná – informace jsou součástí zdrojového kódu – nejen že by musel kantor znát dotýčný programovací jazyk (každý produkt ale může být v jiném PJ), ale v drtivé většině případů by zřejmě došlo i k porušení autorských práv tvůrců materiálu – viz [11]

Zde je zřejmě největší zastoupení produktů – od programů pro tvorbu pomůcek a video souborů až po ucelené systémy.

Tyto programy mohou být „čisté“ – tzn. jedná se pouze o samotný systém, bez dat – ty si uživatel zajišťuje sám. Důvodem využití čistého systému mohou být specifická data učiva (výrobce není schopen taková data zajistit) nebo finance – data mají obvykle několikanásobně vyšší hodnotu než systém - ten je univerzální. Viz [10]. Toto řešení je však obvykle ochuzeno, neboť data nezpracovává tým pracovníků, ale většinou jeden.

Uživatel může ale také získat kompletní systém – program včetně výukové látky. To je nejideálnější možnost a pro atraktivní témata i finančně velice příznivá – např. [12] . Za standart v oblasti e-learningu je kromě jiných brána i firma Langmaster – angažuje se nejen ve školství, ale i ve tvorbě výukových programů pro širokou veřejnost.

1.2 Hardware

PC

Bez něj je (zatím) nemožné e-learning provádět. S ohledem na používané technologie není potřeba výkonných strojů, plně dostačuje PC pro kancelářské práce (obr. 1). Minimální konfigurace a ceny viz Tab. 1.



Obr. 1: PC sestava (MiniPC).

Zdroj: www.GSM4YOU.cz

Tab. 1: Cena minimální konfigurace PC k 1.5.2007 (vše bez DPH)

Název součásti	Popis	Cena
Procesor	AMD 3500+, AM2, 64bit	1 650 Kč
Paměť	512 MB, DDR2, 667 MHz	530 Kč
Základní deska	MSI Sc AM2	1 319 Kč
PC skříň	KME 2362, zdroj 400W	872 Kč
Pevný disk	Seagate 160GB, 7200rpm, 8MB, SATA	1 324 Kč
DVD mechanika	Lite-on LH-18A1P	560 Kč
Monitor	17" AL 1716A	3 119 Kč
Klávesnice	Gembird CZ KB 7400	80 Kč
Myš	Genius Netscroll 110	83 Kč
Reproduktory	Sluchátka s mikrofonem	110 Kč
	Celkem	9 647 Kč

Datový projektor

Není určen přímo pro studenty, ale spíše jako vybavení učebny. Ceny projektorů se pohybují od 15.000,- u základních modelů až po cca 200.000,- u výkonných profesionálních modelů. Při nákupu je potřeba zohlednit především:

I. Rozlišení

Existuje jich několik, označují se buď číselnou hodnotou (např. 800X600) nebo X VGA. Detaily viz tabulka č.2, zdroj

Tab. 2: Rozlišení dataprojektorů

Standart	Popis	Rozlišení	Počet barev
VGA	Video Graphics Array	640 x 480	256
XGA	Extended Graphics Array	1024 x 768	16.7 mil.
SXGA	Super Extended Graphics	1280 x 1024	16.7 mil.
UXGA	Ultra Extended Graphics	1600 x 1200	16.7 mil.

II. Svítivost

„Nejslabší“ projektory mají svítivost 1500 luxů, přičemž minimum je alespoň 2000 luxů. Platí, že čím vyšší svítivost, tím lépe je text čitelný při jasném světle (učebna zalitá poledním sluncem). Při svítivosti 1500 luxů je text za přímého slunečního světla nečitelný, obrázky splývají apod.

U většiny modelů lze přepínat mezi letním a zimním provozem. Zde jde jen o to, že – teoreticky – v zimě není tolik světla a není tak silné, takže žárovka projektoru nemusí svítit tak silně. Naopak v létě je slunce silné, takže je nutné zvýšit svítivost. Rozdíl mezi letním a zimním provozem je obvykle mezi 500 a 1000 luxy. Přestože se zimním provozem se prodlužuje životnost žárovky, většina uživatelů stejně používá pouze letní provoz.

III. Rozměry

Ty je potřeba zohlednit pouze v případě, že už při nákupu počítáme s přenášením projektoru. Mobilní projektory dosahují rozměrů formátu A4 (210x297 mm) a hmotnosti menší než 2kg. Velké modely určené pro statické umístění už jsou poněkud větší – až 80kg při rozměrech 620x385x815 mm (Roadster S+20K od firmy CHRISTIE – něco jako BMW mezi auty)

IV. Chlazení

Používá se buď pasivní (pasivní chladič bez ventilátoru – méně účinné, ale naprosto nehlučné), nebo aktivní (chladič s ventilátorem – účinnost podstatně vyšší, ovšem hlučnost může dosahovat až 60 dB). Pro velké prostory, jako je přednášková hala, je typ chlazení nedůležitý, ovšem v menší učebně může dosahovat až nepříjemné úrovně – takže ohrožuje pracovní hygienu (ergonomii) a snižuje výkonnost přítomných.



Obr. 2: Datový projektor. Zdroj:
www.christiedigital.com

Interaktivní tabule

Uzavírá skupinu základního e-learningového HW. Interaktivní tabule je taková tabule, která je propojena s pc. Obvykle je potřeba spojit PC s tabulí a dataprojektorem – z tabule jdou informace do pc, které je zpracuje a na jejich základě zobrazí pomocí projektoru případné změny na tabuli, kde je uvidí uživatel a může na ně reagovat...Interaktivní tabule tedy funguje jako nadstavba PC pro vizualizaci. Má význam nejen pro zobrazování statických informací, kde je spíše na škodu – dataprojektor musí být seřízen tak, aby osvětloval přesně plochu tabule (ačkoliv může osvětlit menší plochu), takže text je zbytečně malý. Tabule jsou ale nejvhodnější především pro ukázky různých postupů (kdy přednášející pracuje na tabuli stejně jako na PC) nebo se speciálními programy pro interakci – např. lze nakreslit základní obrázek nástroje a studenti dokreslují úhly, řezné síly apod.

Tabule mohou být:

- dotykové – pod povrchem elastické plochy jsou uloženy tlakové senzory, takže tabuli lze ovládat i prstem. Nevýhodou je relativně nízká životnost senzorů a také to, že prst při delší práci bolí... . Na první pohled jsou dotykové tabule nerozeznatelné od Smartboardů.
- magnetické, elektromagnetické – tzv SMARTboardy – v tabuli jsou uložena elektromagnetická čidla, která reagují na změnu mag. pole. Té dosahujeme pomocí

speciálních tužek – lze je přirovnat ke stylusu dotykového displeje (obr. 3.).



Obr. 3: Stylusy pro interaktivní tabule. Zdroj: www.AVmedia.cz

Smartboardy lze rozdělit:

- na tabule s přední projekcí (obr. 4) - tabule je uchycena na zdi, před ní je datový projektor. Promítá se na stejnou plochu, kterou vidí i uživatel.



Obr. 4: Interaktivní tabule s přední projekcí. Zdroj: AVmedia

- Tabule se zadní projekcí (obr. 5) - projektor svítí na zadní stranu, uživatel vidí průsvit. Spíše pro mobilní zařízení.



Obr. 5: Interaktivní tabule se zpětnou projekcí. Zdroj: AVmedia

- e-beam – do rohu běžné bílé tabule se připevní úhelník (obr. 6), který v sobě má dva senzory (obvykle rádiové), které hlídají polohu stylusu na tabuli. Stylusy jsou barevné (každý má jinak naprogramovaný čip).



Obr. 6: E-beam. Zdroj: E-beam.com

- plazmové displeje – jedná se v podstatě o dotkový lcd displej. Ten lze použít buď jako tablet (a pak se promítá na běžné plátno, displej má malé rozměry – do 17“), nebo jako interaktivní tabulí (nahrazuje samotnou interaktivní tabuli s dataprojektorem) – obr. 7.



Obr. 7: Plazmová interaktivní plocha. Zdroj: Image.cz

2 Výběr optimálního řešení

Správně by měla být tato kapitola nadepsána Výběr „optimálního“ řešení, protože nejlepší řešení neexistuje (resp. neexistuje univerzální nejlepší řešení). Již z výčtu existujících řešení je patrné, že každé má své výhody i nevýhody, se kterými musí případný tvůrce počítat.

Budeme-li se věnovat našemu konkrétnímu případu, tzn. multimediální encyklopedii obrábění pro studenty střední školy, vyplývají z tohoto úkolu pro jednotlivá řešení různé

výhody a omezení:

2.1 Software

● ***Email*** – nepřináší žádný pokrok, rozdíl proti klasické výuce by byl pouze ve změně přenosového média. Komunikace se přenese z fyzického kontaktu na vzkazy elektronické pošty. Studenti by své odpovědi museli zapisovat do PC a vzhledem k tomu, že 98% neumí psát všemi deseti, resp. má problém napsat obsáhlejší dokument (obsáhlým dokumentem mám na mysli více než jednu normo stranu textu), docházelo by k prodlevám, větší časové náročnosti při studiu a většinu studentů by to spíše demotivovalo. Pro lektora vznikají omezení např. při zkoušení – nelze ověřit, kdo vypracoval úkol, jak dlouho mu trval, Výhodou je snad jen možnost práce na jakékoliv platformě OS a nízké nároky na klientský počítač.

Také vzhledem k tomu, že studenti mají možnost získat podkladové materiály v elektronické podobě od učitele, postrádá tato metoda smyslu.

● ***HTML, XHTML*** – technologie přímo předurčená pro e-learning, je relativně jednoduchá na zvládnutí, nezatěžuje příliš PC, objemy přenášených dat jsou malé. Silným nástrojem je jednoduché vyhledávání v HTML dokumentech.

Bohužel ani jedna z technologií dnes neobstojí samostatně a musí se propojovat s dalšími – např. SQL databázemi pro možnosti vyhledávání v textu, CSS pro pokročilejší formátování, PHP a Java applety pro další funkce. Na tyto funkce je pak vázáno další programové vybavení klientského PC – v prohlížečích musí být doinstalovávány další pluginy, Java, Direct X,

Z toho vyplývá, že pokud má být plně využita HTML technologie, musí autor ovládat nejen ji, ale i další (navazující) obory.

● ***Databáze*** – skýtá velké možnosti pro získávání, uchovávání a zpracování dat. Bohužel možnosti databází jsou silně omezené v oblasti multimédií (přestože lze do databáze zařadit zvuk i videosekvence, jde o poměrně složitý postup). Další faktor, který snižuje kvalitu databází pro naši encyklopedii může představovat problém při přenosu na jinou platformu OS.

Čtenář znalý alespoň základní problematiky databází si již určitě uvědomil další omezení – aby databázový systém fungoval, potřebuje dvojici PC; server-klient. Nezáleží na tom, zda je klient zároveň serverem (v tomto případě je vyžadována přítomnost např. My SQL serveru, což je u běžného uživatele téměř neřešitelný problém), nebo se k serveru připojuje

na síť. Pak je ale vyžadována přítomnost dalšího stroje, na kterém je server podporován – zvyšují se finanční náklady. Z vlastní zkušenosti vím, že správci školních sítí se zuby nehty brání spouštění jakékoliv jiné služby na „svém“ serveru – ať už z neznalosti, strachu o bezpečnost sítě nebo neochoty věnovat svůj čas něčemu navíc.

● Programy pro e-learning

- obecné programy – přestože jsou nejrozšířenější, nelze využít zpětné vazby, také interaktivita je minimální
- programy pro tvorbu e-learningových materiálů – vyžadují sice znalosti dotyčných programů, odměnou je však vysoká interaktivita se studentem, nechybí ani zpětná vazba. Rozšiřitelnost je také bezproblémová.
- e-learningové programy – i když jsou kvalitní, jsou bohužel drahé a nelze je upravovat. Pro tvorbu je potřeba dobré znalosti vyššího programovacího jazyka.

2.2 Hardware

Závisí na vybrané SW technologii, přitom je úzce spojen s finanční stránkou – cena za zřízení a provoz musí být co nejnižší – jak z hlediska provozovatele (školy), tak z hlediska uživatele (studenta).

V ideálním případě by náklady na HW měly být nulové – využijeme stávajících možností. Zde vycházím z předpokladu, že téměř všichni studenti vlastní PC, nebo k němu mají alespoň přístup. Také každá škola je dnes vybavena alespoň základní výpočetní technikou (PC, datové projektory).

2.3 Shrnutí

Zkombinujeme-li jednotlivé faktory, našemu záměru zřejmě vyhovuje použití technologie HTML nebo programů pro e-learning.

HTML vyžaduje server a stálé připojení uživatele k internetu. Z vlastní zkušenosti vím, že i když žáci mají PC nebo přístup k němu, s připojením je to horší. Tato technologie je pro nás tedy nevyhovující.

Zbývají programy pro e-learning. Obecné nástroje nepřicházejí v úvahu – jednak postrádá toto řešení interaktivitu, navíc studenti již mají možnost získat od učitele podklady k výuce

v elektronické podobě. E-learningové programy nepřicházejí v úvahu – předně neexistují (tento fakt byl vlastně impulzem k vytvoření této práce). Je možno si takový program objednat, ale všichni známe finanční situaci našich škol, takže se asi shodneme, že cca 30.000,- Kč za jeden elektronický kurz škola nezaplatí, protože na něj prostě nemá. Zbývá tedy poslední možnost – využití nějakého programu určeného pro tvorbu e-learningu.

Vzhledem k mému zaměření padla volba na produkt firmy Adobe – Flash. Práce začala vznikat ve verzi 8 Pro, ale protože během zpracování vyšla verze CS3, celá práce byla převedena do této nové verze a dokončena v ní.

Díky využití Flash playeru (zmínka o něm je v kapitole 1.1.1 SW řešení) je zajištěno spuštění a provoz výsledné aplikace na každém PC. Vzhledem k technologii, kterou Flash využívá, nebude ani velikost prezentace nijak závratná (pohybuje se kolem 100 MB). Další obrovskou výhodou je možnost tvořit prezentaci po etapách, v závislosti na úrovni znalostí řešitele:

- I. etapa – tvora prostředí, navigace a zadání informací pro výuku, zprovoznění encyklopedie
- II. etapa – pomocí AS provázání vnitřních struktur, zprovoznění funkce vyhledávání, časových funkcí
- III. etapa – pomocí WSDL a AS přidat provázání s případným serverem

Tedy teprve ve třetí fázi by bylo možné (nikoliv nutné) využívat služeb serveru. Tento server by byl provozován na serveru nabízejícím své služby zdarma např. [13].

Finanční náklady jsou popsány v tabulce 3. (práce autora není počítána – prezentace je řešena v rámci DP), cena SW je samozřejmě pouze pro školní licence, v tomto případě při nákupu jediné licence.

Položka	Cena
Nákup Flash CS3	6.589,-
Práce autora	0,-
Další SW (Photoshop, Premiere)	5.320,-
Celkem	11.909,-

Tab. 3: Kalkulace finančních nákladů (ceny jsou uvedeny bez DPH).

FLASH (PŘEDSTAVENÍ PROGRAMU)

Jedná se o multimediální aplikaci, která umožňuje programátorům tvořit, využívat a sdružovat grafické, textové, zvukové a video objekty, vytvářet animace, integrovat je a distribuovat je. Distribuce lze zajistit pro téměř libovolný operační systém (od libovolné distribuce Linuxu, přes Windows až po Solaris) a zařízení (PC, mobilní telefon, PDA, ...).

Díky tomu, že na většině PC je dnes již automaticky instalován Flash player (speciální přehrávač .swf souborů – tedy produktů Flashe), není problém s rozšiřováním výtvorů. A protože FLASH využívá vektorovou grafiku (vysvětleno v kapitole 2.3 Kreslení), je přímo předurčen pro šíření on-line.

Součástí programu je i vcelku obsáhlá knihovna předpřipravených prvků: tlačítka, grafické symboly, datové komponenty, základy formulářů.

Dalším důležitým prvkem je i prostředí pro tvorbu vlastních aplikací. Flash využívá vlastní programovací jazyk Action Script (dále jen AS) určený nejen pro vytváření vztahů mezi jednotlivými objekty, ale lze jej využít i pro ovládání vlastností objektů – vytváříme tak „dynamické“ objekty. AS prošel rozsáhlým vývojem, takže nejnovější verze nese označení AS 3.0 a lze pomocí něj ovládat i tzv. 3D zobrazování. Záměrně píšeme takzvané, protože Flash neumí plně 3D, pouze jej simuluje.

1 Filozofie produktu

Celý systém je založen na časové ose. Ta obsahuje snímky, do kterých vkládáme jednotlivé prvky (grafiku, text,...). Při přehrávání se zobrazují jednotlivé snímky podobně, jako když se přehrávají políčka klasického filmu. Flash má tu výhodu, že můžeme (a měli bychom) využívat vrstvy, které jsou na sobě nezávislé. V praxi to tedy znamená, že použijeme jednu vrstvu na zobrazení grafiky, v ní se např. nachází barevný obdélník. Ten se v dalších snímcích posunuje dál a pro uživatele tak vzniká dojem pohybu. Nad tuto vrstvu pak umístíme další vrstvu, ve které se nachází text. Tyto vrstvy mají společnou pouze základní časovou osu, takže obdélník se pohybuje nezávisle na textu.

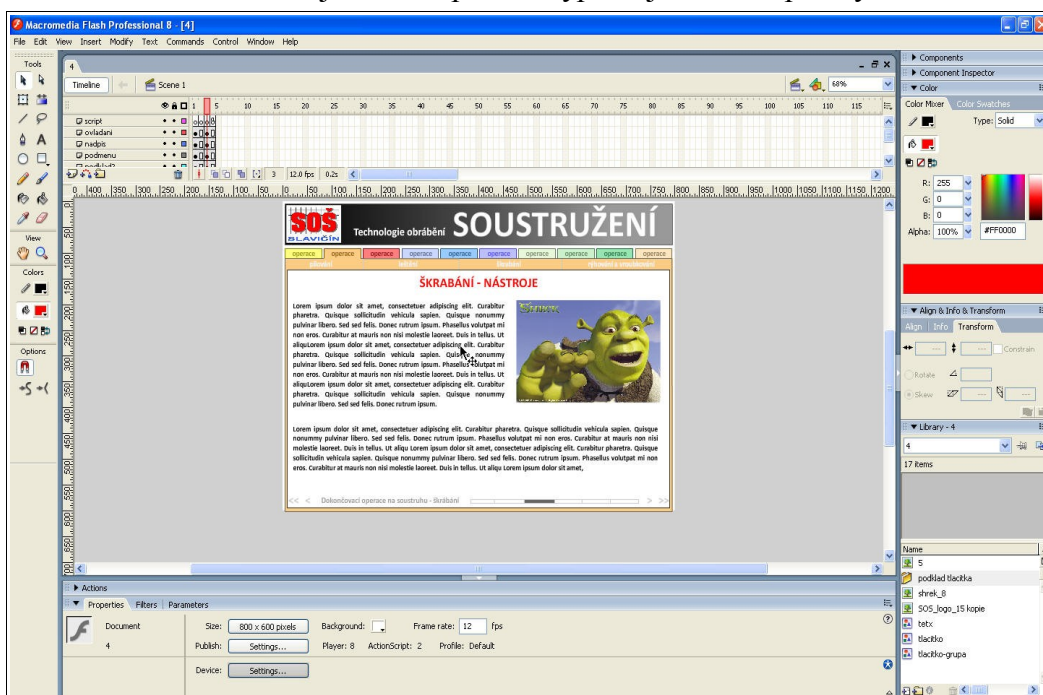
Pohyb, nebo jakoukoliv změnu lze vytvořit buď pomocí změny „políčko po políčku“ (ručně zadáváme stav objektu po jednotlivých políčkách), nebo pomocí tzv. „tweenu“ (nastavíme počáteční a koncové vlastnosti a počítač sám propočítá přechody), nebo pomocí AS (vše je třeba naprogramovat).

Další „vymožeností“ je použití knihoven. Prvek můžeme vytvořit pouze jednou a pak jej opakovaně využívat. Výhoda pak spočívá ve snížení velikosti souboru – prvek je v něm uložen pouze jednou a my pak při práci vyvoláváme pouze tzv. instance symbolů (je to podobné jako např. zástupci programů ve Windows – program je skryt v adresářové struktuře a na ploše je pouze odkaz na něj). Při změně symbolu v knihovně se automaticky změní všechny instance, takže odpadá složité vyhledávání a výměna všech dílů. Instance lze pojmenovat a pak je volat pomocí AS. Míra úspory místa samozřejmě závisí na počtu instancí – pokud prvek z knihovny použijeme pouze jednou, úspora bude minimální (neřku-li nulová). Čím více instancí v souboru využijeme, tím větší bude úspora. Více o symbolech v kapitole 2.4 Symboly.

Poznámka: Protože většina projektu je vytvořena ve Flashi 8 Pro, budu zde popisovat právě tuto verzi, přestože aktuální verze je Flash CS3.

2 Pracovní prostředí

Pracovní prostředí (obr. .) má několik částí (panelů), jejich polohu, velikost a další vlastnosti lze měnit. Samozřejmě lze zapínat a vypínat jednotlivé panely.



Obr. 8: Pracovní plocha

Po kompletním nastavení celého prostředí (velikost a poloha nástrojů, paletek,..) lze svůj profil uložit, takže je možno jej přenášet mezi počítači i instalacemi – Windows / Workspace Layout / Save current.

Podobně lze měnit a ukládat klávesové své zkratky – Edit / Keyboard shortcuts.

2.1 Paletky nástrojů

Jedná se o panely ve kterých jsou sdruženy nastavení týkající se určitých vlastností (výčet paletek následuje). Tyto paletky bývaly v dřívějších verzích (MX aj.) plovoucí, v novějších verzích už jsou standardně umístěny k pravému okraji pracovní plochy. Tyto paletky lze také sdružovat do skupin – typickou skupinou je např. Zarovnání (Align) – Transformace (Transform) nebo Barvy (Colors) – Barevné vzorníky (Color swatches) atd.

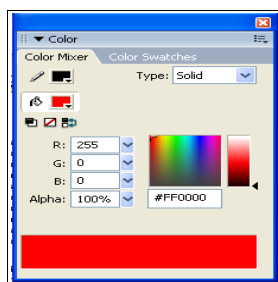
Po upravení pracovní plochy k obrazu svému lze toto nastavení samozřejmě uložit a pak přenášet na jiné stroje nebo instalace.

Seznam Paletek:

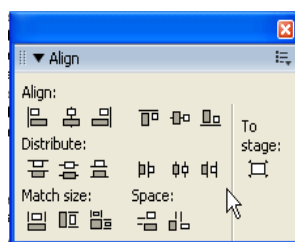
- Akce (Actions) – editor AS, lze zadávat buď ručně, pomocí chování (Behaviors) nebo pomocí asistenta – AS je blíže popsán ve zvláštní kapitole.
- Barevné vzorky (Color swatches) – paletku základních barev lze rozšiřovat o vlastní barvy (a vytvořit si tak vlastní vzorník zvláštních barev, hotových přechodů apod.)
- Barvy (Color nebo Color mixer) – nastavení barev, barevných výplní, přechodů, bitmap... - obr. 9.
- Časová osa (Time line) – paletka pro ovládání osy, taktéž má vlastní kapitolu
- Hlavní (Main) – základní nástroje pro práci dokumenty (Nový, Uložit, ...)
- Info (Info) – zobrazuje info o poloze kurzoru, vlastnostech vybraného objektu jako je barva velikost, poloha – obr..
- Knihovna (Library) – jedna z nejdůležitějších paletek. Zde se ukládají všechny objekty a komponenty použité v dokumentu. Pokud chceme nějaký prvek z knihovny použít, přetáhneme jej na plochu, čímž vznikne jeho instance.
- Knihovna přednastavených prvků (Common libraries) – obsahuje tři typy knihoven přednastavených prvků – zvuků, tlačítek, komponent.
- Komponenty (Components) – knihovna předem připravených komponent – tlačítka, části formulářů, ovládací prvky apod. Vybranou komponentu prostě přetáhneme z knihovny komponent do scény. Tím dojde k vytvoření instance, se kterou pak pracujeme (více viz kapitola Symboly). Bohužel předpřipravené komponenty se

hůře editují, dají se však sehnat na internetu - [14] (placené i volně ke stažení).

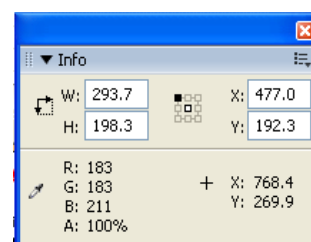
- Ladění (Debugger) – panel pro ladění skriptu, více v kapitole Action script.
- Nástroje (Tools) – základní nástroje pro kreslení, o této paletce pojednává samostatná kapitola)
- Odpovědi (Answers) – v podstatě rychlé odkazy nápovědy a tipy
- Ovladač (Controller) – ovládání přehrávání animace
- Parametry komponent (Component parameters) – zobrazení vlastností vybraných komponent podobně jako v panelu Vlastnosti.
- Prohlížeč snímků (Movie explorer) – jednotlivé objekty obsažené ve scéně jsou seřazeny do kategorií (text, tlačítka, symboly, ...). Zobrazení kategorií lze filtrovat.
- Příslušnost (Accessibility) – nastavení dostupnosti dokumentu, především čitelnost.
- Reference (Reference) – panel nápovědy k Action scriptu, seznam všech funkcí AS.
- Scéna (Scene) – práce se scénami – tvorba nových, spravování stávajících.
- Status (Status) – informační řádek ve spodní části programu
- Transformace (Transform) – změna velikosti objektů (v měřítku nebo zvlášť v ose X nebo Y), rotace, zkosení
- Vlastnosti (Properties) – obsah se mění podle vybraných objektů nebo nástrojů, viz samostatná kapitola
- Výstup (Output) – Výpis vlastností dokumentu, zobrazuje se pouze při exportu do distribučních souborů jako je .swf, .html, atd.
- Zarovnání (Align) – zarovnání objektu vzhledem k hranicím scény, zarovnání více objektů vůči sobě, rovnoměrné rozmístění objektů, ... - obr.10 .



Obr. 9: Paletka Barvy



Obr. 10: Paletka zarovnání

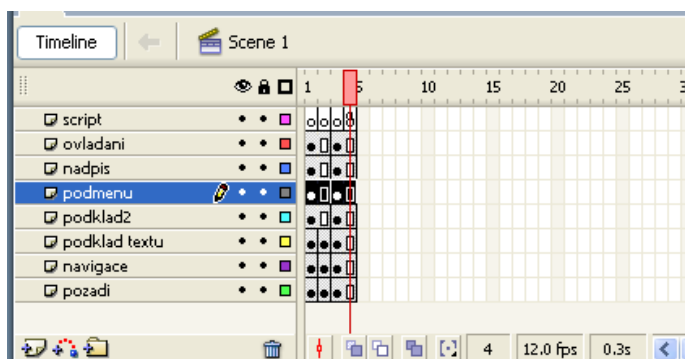


Obr. 11: Paletka Info

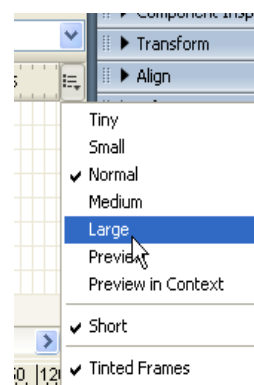
Z těchto paletek jsou následující tři zřejmě nejdůležitější a práce ve Flashi by bez nich byla nemožná.

2.2 Časová osa (Timeline)

Zobrazuje jednotlivé snímky. Lze měnit velikost snímků – zvětšovat, zmenšovat, měnit jejich výšku tak, aby bylo jejich zobrazení optimální (viz obr. 13)



Obr. 12: Detail časové osy



Obr. 13: Velikosti snímků

V levé části osy je seznam vrstev, pod nimi tlačítka pro ovládání vrstev (viz obr. 12 – Nová vrstva, Guide (nastavení cesty pohybu), Nová složka vrstev a nakonec Koš (pro odstranění vrstev i složek vrstev). Nad seznamem vrstev jsou další tlačítka – zapínání/skrývání vrstev (zavazí-li nám vrchní vrstva ve výhledu), zamykání/odmykání vrstev (pokud nechceme při práci s vrstvou omylem měnit i jiné vrstvy) a poslední tlačítko pro plné zobrazení/obrysové zobrazení (vhodné pro složité scény, náročná zobrazení apod – sníží se tak nárok na výkon pc a scéna je dříve vykreslena). Použitím těchto tlačítek zapínáme/vypínáme vlastnosti všech vrstev a složek vrstev. Pokud bychom chtěli měnit vlastnosti konkrétních vrstev, musíme použít tlačítka vedle názvu vrstvy (samozřejmě v příslušném sloupci).

Pod časovou osou se nacházejí další ovládací a informační prvky – obr. 12.- ovládaní osy, číslo snímku, na kterém se právě nachází jezdec, FPS scény, a čas, ve kterém se přehraje aktuální snímek (ten je přepočítáván z FPS a čísla snímku).

Přes pravé tlačítko nebo dvojklikem lze vyvolat další vlastnosti vrstvy – přejmenovat ji, změnit její barvu (ta je ale podstatná pouze při obrysovém zobrazení).

Na časovou osu klademe jednotlivé snímky. Ty mohou být:

1. Klíčový snímek (keyframe) – klíčový snímek. Takový snímek, ve kterém se nachází nový obsah (nebo výrazně jiný). Typicky je to první a poslední poloha animace

apod.

2. Snímek (frame) – běžný snímek, který má stejný nebo velmi podobný obsah jako předchozí klíčový snímek.

Příklad použití snímků

Potřebuji vytvořit animaci modrého obdélníku, který se přesune z levého horního rohu do pravého spodního. Postup je takový že na prázdné časové ose vytvoříme ve snímku č.1 klíčový snímek, do něj z knihovny vložíme modrý čtverec do levého horního rohu scény. Podle požadované doby trvání a FPS určíme poslední snímek animace. Do něj vložíme další klíčový snímek, ve kterém bude modrý čtverec v pravé dolní části scény. V tomto okamžiku vznikly mezi oběma klíčovými snímky normální snímky – jako bychom první klíčový snímek roztáhli na více snímků.

Přejedeme-li nyní jezdce časové osy přes naplněné snímky, zjistíme, že čtverec je v levém horním rohu po celou dobu a teprve na posledním snímku se skokem přesune na pozici vpravo dole.

V tomto okamžiku bychom pokračovali animací (viz 2.6.1 Motion tweening).

2.3 Scéna

Je základem všech dokumentů Flashe. V podstatě se jedná o prázdný prostor, do kterého vkládáme jednotlivé prvky. Opět lze využít analogii s filmem – scéna sdružuje určité situace podle vlastností, podmínek, charakteru (viz[15]). V každém souboru je alespoň jedna, scény jsou navzájem nezávislé, každá má vlastní časovou osu.

Samotná scéna není pro práci důležitá z hlediska tvorby ale hlavně z hlediska uspořádání. Obvyklým příkladem použití scén je často používaná složitější (resp. obsáhlejší) prezentace. V ní použijeme první scénu pro úvod, druhou pro první kapitolu, třetí pro navigaci apod.

Pořadí scén lze určit jednoduchými příkazy AS, pokud je však neurčíme, budou se zobrazovat ve stejném pořadí, v jakém jsou seřazeny v paletce Scene. Pokud nebudou scénu ukončeny příkazem

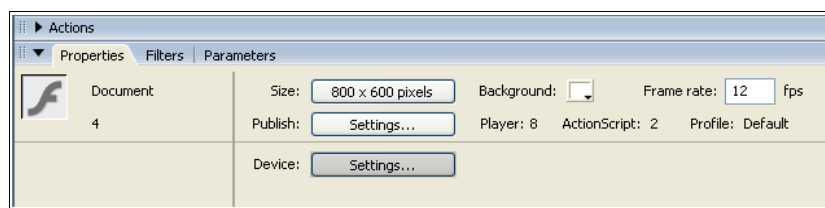
```
stop ();
```

budou se opakovaně přehrávat stále dokola.

2.4 Vlastnosti (Properties)

Obvykle ve spodní části prostoru, spojen s panely Akce (Actions) a Parametry (Parameters). Viz Vlastnosti, které lze v tomto panelu nastavovat, se mění podle toho, jaký objekt máme vybraný, případně s jakým nástrojem zrovna pracujeme.

Např. po kliknutí na pracovní plochu se zobrazí vlastnosti celé scény – barva pozadí, FPS, velikost scény, ... - obr. 14.



Obr. 14: Panel vlastností

Pro symboly lze používat položky jméno instance, pro text, čáry a výplně barvy, u textu barvy, velikost, typ textového pole apod.

2.5 Nástroje (Tools)

Paleta základních nástrojů pro práci. Opět ji lze měnit k obrazu svému – přidávat nebo ubírat ikony nástrojů, spojovat je do skupin atd. Po zapnutí některého z nástrojů se změní panel Vlastností (Properties) i spodní část lišty nástrojů – Nastavení (Options) tak, aby zobrazovaly adekvátní vlastnosti nástroje.

V základním nastavení jsou zobrazeny tyto nástroje - jelikož neexistuje česká verze, uvádím i anglické názvy tlačítek a nástrojů – viz obr. 23:

Výběr - Arrow – pro výběr objektů, přesouvání, editaci. Při stisknutí nad koncovým bodem čáry nebo rohem objektu s ním mohou pohybovat, kliknutím nad libovolným bodem čáry můžeme měnit zakřivení (i u úseček).

Jedním kliknutím vyberu jediný prvek, dvojklikem na čáru vyberu všechny navazující čáry se stejnými vlastnostmi, dvojklikem na prvek vyberu jak jeho výplň, tak i obrys. Dvojklikem na symbol přejdu do režimu editace symbolu přímo ve scéně.

Podvýběr – Subselection – pouze pro editaci křivek a koncových bodů. Při kliknutí na čáru se zvýrazní významné body křivky, se kterými pak můžeme pohybovat. Také můžeme měnit směr a velikost zaoblení v těchto bodech

Úsečka – Line – základní kreslicí nástroj, pro kreslení rovných čar (poté lze tvarovat pomocí nástroje výběr). Vlastnosti čar nastavujeme v panelu Vlastnosti (viz níže)

Laso – Lasso – slouží k výběru nepravidelných tvarů, částí objektů. Může pracovat v režimu od ruky (nakreslím libovolně tvarovanou oblast), podle barvy (po kliknutí se označí souvislé plochy se zadanou barvou) a polygon (značení pravidelných tvarů pomocí úseček, každé kliknutí vytvoří nový rohový bod spojený s předchozím úsečkou. Tvar se uzavře poklepnutím na počáteční bod).

Pero – Pen – pro tvorbu vektorových křivek pomocí klíčových bodů. Nejprve se umístí bod, pak se nastaví směr a velikost zakřivení.

Text – Text – jak název napovídá, nástroj slouží ke vkládání textu, vlastnosti se nastavují v panelu Vlastnosti. Má podobné funkce jako v textových editorech, ovšem ne tak pokročilé. Více v kapitole 2.4 Text.

Ovál – Oval – kreslení kruhových a oválných objektů. Po zapnutí nástroje se objeví nastavení čáry a výplně. Kreslíme-li ovál se zmáčknutou klávesou Shift, automaticky se kreslí kružnice, při zmáčknuté klávese Alt kreslíme ovál od středu (standardně od levé horní části)

Obdélník – Rectangle – kreslení pravoúhlých čtyřúhelníků. Stejně jako u nástroje Ovál se objeví vlastnosti čar a výplně. Podobně fungují i klávesy Shift (kreslíme čtverce) a Alt (kreslení od středu).

Tužka – Pencil – klasické kreslení od ruky. V nastavení lze vybírat mezi režimem zaoblování (čára bude plynulá bez ostrých rohů), zarovnávání (i křivé čáry budou nahrazeny rovnými), inkoust (čára nebude upravena a zůstane tak, jak je).

Štětce – Brush – pracuje podobně jako tužka, ale čára vytvořená štětcem má vlastní obrys a výplň – lze tedy kreslit čáry s lineárními přechody, bitmapovou výplní, Křivka se po nakreslení chová jako objekt, tvar lze měnit nástrojem podvýběr. Při práci lze v panelu nastavení vybírat z následujících atributů: Tvar štětce (kulatý, oválný, čtvercový, lichoběžníkový,...), Tloušťka čáry, Mód štětce (normál, kreslení výplně, kreslení kolem, ...

Volná transformace – Free Transformation – po označení objektu se vyznačí jeho obrys včetně klíčových bodů. Při tažení těchto bodů lze měnit výšku, šířku, úhly, velikost, natočení a další vlastnosti objektu spojené s jeho polohou. Při zmáčknuté klávese Ctrl lze

editovat každý bod zvlášť.

Změna výplně – Fill transform – nástroj pro změnu výplně – otočení, posun, transformaci. Má význam pouze u výplní tvořených bitmapou nebo gradientním přechodem.

Inkoust – Ink Bottle – při označení ohraničení tímto nástrojem lze měnit nastavení čar (ohraničení). Vybereme tento nástroj, nastavíme požadované vlastnosti a označíme čáry, u kterých požadujeme nově nastavené vlastnosti.

Výplň – Paint Bucket – změna nebo tvorba nových výplní všech druhů (barva, přechod, bitmapa,..). Lze nastavovat další vlastnosti – úplně uzavřené objekty, částečně uzavřené objekty, přesnost značení.

Kapátko – Dropper – jak už název napovídá, slouží kapátko k „nabrání“ požadované barvy – ta se pak nastaví do okénka barvy výplně nebo přímo do předem vybrané výplně, kterou tak nahradí.

Guma – Eraser – slouží k mazání čar, výplní (i jejich částí). V panelu nastavení lze opět vybírat z několika režimů: normál, výplně, čáry, vybrané výplně, uvnitř a v neposlední řadě i barva – po zapnutí tohoto režimu dojde k vymazání souvislé plochy obarvené vyznačenou barvou.

Ruka – Hand – není určena přímo pro kreslení, umožňuje pohybovat pracovní plochou. Tento nástroj má samozřejmě význam pouze při vysokém zvětšení scény, takže není celá vidět. V některých programech bývá tato funkce označována jako „Pan“.

Přiblížení – Zoom – kromě nastavení přímé hodnoty přiblížení v časové ose lze využít tento nástroj. Po zapnutí funguje jako zvětšení, při podržení klávesy Alt naopak jako zmenšení. Pro funkci přiblížení je nejvhodnější používat klávesovou zkratku (M, Z).

3 Kreslení

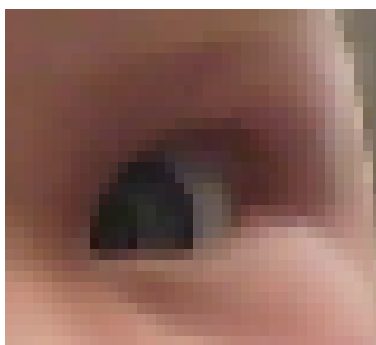
Kreslení, resp. práce při kreslení je založena na faktu, že Flash využívá **vektorové grafiky** (opakem je grafika bitmapová, kterou lze sice také využít, ale pokud možno v omezené míře – mimo jiné neúměrně zvyšuje velikost výsledného souboru a časy pro načtení a zobrazení těchto bitmap). Pro úplnost uvádím krátké definice pojmu vektorová a bitmapová grafika:

Bitmapové grafika – někdy také bitmapa, uchovává obraz tak, že jej rozdělí na konečný počet bodů (tzv. pixelů) a každému bodu pak přidělí vlastnosti jako je barva, jas aj. Celý

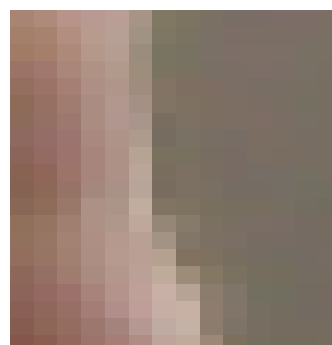
obraz je tedy poskládán z velkého množství bodů. Tento způsob ukládání je vhodný pro komprimaci, a pro obrazy s plynulými přechody jako u digitální fotografie. Bohužel má bitmapa svá omezení – i při ztrátových kompresích (jpg, gif,...) je velikost souboru stále relativně velká, takže se příliš nehodí pro on-line distribuci. Důležitější omezení však poznáme, pokud budeme bitmapový obrázek zvětšovat: pokud dosáhneme určitého zvětšení (běžně 300 – 400% podle obsahu obrázku), lze rozeznat na barevných přechodech určité nesrovnalosti a při dalším zvětšování (nad 500%) můžeme rozpoznávat jednotlivé body (viz obr 15, 16, 17).



Obr. 15: Bitmapa - 100%



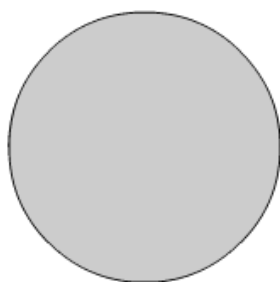
Obr. 16: Bitmapa - 400%



Obr. 17: Bitmapa - 650%

Bitmapa tedy není vhodná tam, kde můžeme počítat se zvětšováním – např. u digitálních fotoaparátů má rozlišení (tj. počet pixelů) vliv na maximální velikost fotografie. Obecně platí, že čím větší rozlišení, tím větší může být vytištěná fotografie – viz [16].

Vektorová grafika – vynikající pro „jednodušší“ grafiku, ve které jsou jasné a ostré přechody a ohraničení (přesně takovou grafiku využíváme právě ve Flashi). Popis obrázku je pak tvořen matematickým popisem jednotlivých křivek v prostoru, takže při případné změně měřítka (především při zvětšování) dojde k přepočítání křivek a nedochází ke ztrátám ostrosti apod- viz obr 18,19. Zoubkování v obr. 19 je způsobeno tím, že vektorový obrázek je dále zpracováván jako bitmapa.



Obr. 18: Vektor - 100%



Obr. 19: Vektor - 400%

Tato grafika je vynikající pro on-line šíření (malé objemy přenášených dat). Za nevýhodu lze pak považovat nutnost přepočítávání scény při každé změně měřítka – ta ale probíhá na klientském PC, které už je obvykle dostatečně výkonné (viz Tab. 1), takže prodlevy jsou minimální (nepostřehnutelné).

Při použití vektorové grafiky tedy vytváříme objekty, které jsou tvořeny obrysem (čarou), výplní a ohraničením. Navíc je takový objekt plně editovatelný po celý čas práce. Atributy však nejsou povinné – objekt může být tvořen (dle [17]):

- ohraničením s výplní
- výplň bez ohraničení
- ohraničení bez výplně



Obr. 20: Typy objektů (zleva): s obrysem a výplní, s ohraničením bez výplně, pouze výplně a jako poslední je objekt s ohraničením

Objekty lze kreslit několika způsoby:

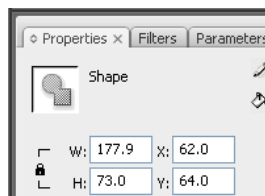
- jednoduché tvary – lze nakreslit pomocí základních nástrojů – ovál, obdélník, tužka. Tyto tvary pak lze deformovat pomocí nástroje výběr, podvýběr a transformace.
- složitější tvary - můžeme vytvořit spojováním nebo odebráním jednoduchých tvarů nebo kreslením a editací křivek.

Kreslení výrazně usnadňuje souřadný systém. Využívá osy X a Y, počátek soustavy (bod $[0;0]$) se nachází v levém horním rohu. Osa X je vodorovná, Y svislá – obr. 21.



Obr. 21: Souřadný systém

V panelu nástrojů pak můžeme zadávat přesné souřadnice klíčových bodů objektů a tím je zaručena požadovaná přesná poloha.



Obr. 22: Souřadnice a velikost objektu - Panel vlastností

3.1 Čáry

Čára je ten nejzákladnější element, který při kreslení využíváme – můžeme vytvářet obrysy, křivky pro určení trajektorie, dělicí elementy apod. Při práci využíváme tři typů čar:

Přímky

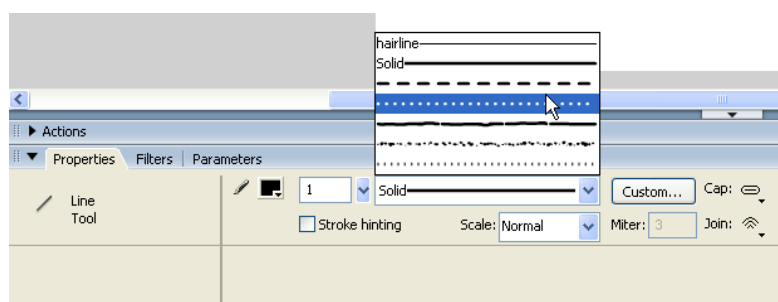
neboli Line tool. Jsou vhodné k vytvoření rovných úseček (lze je dále deformovat – viz níže). Pro spuštění nástroje použijeme buď ikonu Line tool, (obr. 23), nebo klávesovou zkratku „N“. Při kreslení označíme počáteční bod čáry (LMB), tažením a uvolněním určíme koncový bod.



*Obr. 23: Nástroje čar:
Červená - Přímky,
Zelená - Křivky,
Modrá - Tužka,
Fialová - Štětce*

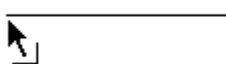
Přímce můžeme nastavovat především tyto vlastnosti (obr. 24):

- barva (z paletky barev)
- síla (jezdcem nebo číselnou hodnotou)
- typ čáry (normální, tenká, přerušovaná,...)
- vlastní nastavení (jde o kombinaci síly a typu čáry)
- zakončení čáry (kulaté, hranaté, žádné)



Obr. 24: Vlastnosti čar - nástroj přímka

Přímky lze editovat pomocí nástroje výběr – pokud umístíme kurzor nad konec čáry (obr25), můžeme je přetáhnout jinam – čára změní délku a směr. Pokud kurzor umístíme kdekoliv nad čáru, můžeme ji zakřivit (obr. 26). Tento způsob je použitelný i na ostatní typy čar.



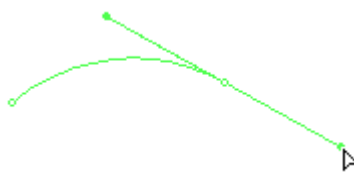
Obr. 25: Změna
koncového bodu čáry
(kurzor)



Obr. 26: Změna zakřivení přímky: tvar
kurzoru, výsledek

Křivky

neboli Pen tool. Spouští se ikonou (obr. 23), nebo klávesovou zkratkou „P“. Nástrojem vytváříme objekty podobné Beziérovým křivkám a to tak, že určíme bod, kterým prochází a pak nastavíme vektor z tohoto bodu (obr. 27). Práce však vyžaduje jisté minimální zkušenosti, jinak je výsledek podstatně odlišný od původního záměru. Nastavujeme stejné atributy jako u přímek.



Obr. 27: Křivka - nastavení směrového vektoru

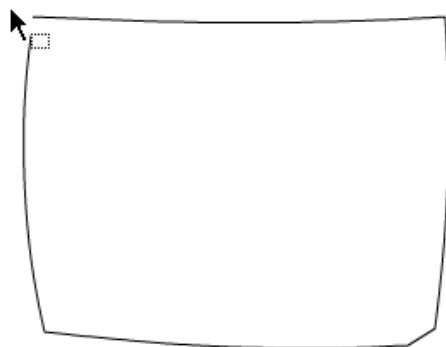
Editace je – kromě nástroje výběr – možná nástrojem podvýběr. Po značení křivky se zvýrazní nejen její klíčové body, ale také jejich směrové vektory (obr. 28), takže můžeme měnit tvar citlivěji. Samozřejmostí je přidávání a ubírání počtu klíčových bodů.



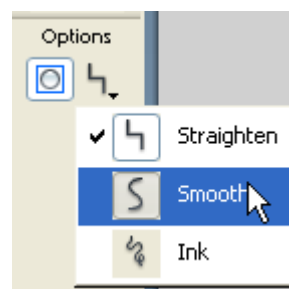
Obr. 28: Editace křivky nástrojem podvýběr

Tužka

neboli Pencil tool. Spouštíme ikonou (obr 23), nebo klávesovou zkratkou „Y“. Používá se ke kreslení čar „od ruky“. Kromě nastavení atributů čáry (viz Přímký), můžeme ještě nastavovat styl (hranatá čára, zaoblená, beze změn – obr. 30) a tzv. „kreslení objektů“ - pokud nakreslíme tvar byť vzdáleně se podobající obdélníku (obr. 29), program jej na obdélník skutečně převede. Totéž samozřejmě platí i pro další obrazce.



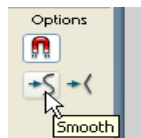
Obr. 29: Tužka - i takový obrazec stačí pro doplnění na čistý obdélník



Obr. 30: Styl čáry - nástroj tužka

Pokud by k PC byl připojen tablet s tlakovou citlivostí, bylo by možno nastavovat ještě tlakovou a směrovou citlivost tabletu přímo v programu.

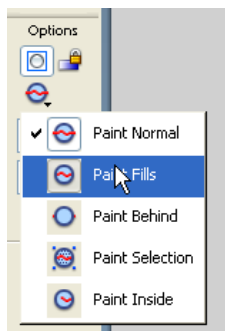
Při editaci lze využít dalších nástrojů – zaoblení nebo zостření čar (obr. 31).



Obr. 31: Nástroje
zaoblení a zостření čar

Štětec

Neboli Brush tool. Spouštíme ikonou (obr. 23), nebo klávesovou zkratkou „B“. Čáry vytvořené štětcem jsou samostatnou kategorií čar - fungují podobně jako čáry „od ruky“, ale navíc lze nastavovat další atributy – především výplň (čára nemusí mít jen jedinou barvu, ale lze využívat i přechodů a výplní – obr. 33) a mód (čára vzniká normálně, v objektu, za objektem, ... - obr. 32).



Obr. 32: Módy
nástroje štětec



Obr. 33: Štětec - vytváří čáry s výplní

Editace nástrojem výběr je jediná možnost, ale měníme pouze obrysy čar, takže se rozhodně nejedná o jednoduchou záležitost.

3.2 Obdélník (Rectangle)

Po zapnutí nástroje Obdélník si v nastavení můžeme okamžitě vybrat výplň a obrys (obr 34).



Obr. 34:

Nástroj

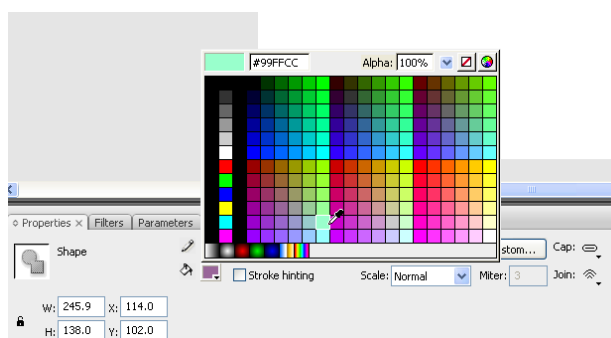
Obdélník

Při kreslení pak stiskneme levé tlačítko (dále LMB) – tím určíme první roh. Při stále stisknutém LMB pohybujeme kurzorem, čímž určujeme protilehlý roh (úhlopříčný).

U obdélníků lze jako u jediného nastavit ještě nastavovat zaoblení.

Chceme-li pak objekt editovat, postupujeme následujícími způsoby – až na výjimky jsou tyto postupy shodné i pro ostatní tvary:

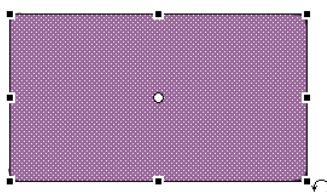
- Barva výplně: klikneme jednou do výplně. Tím je označena a v panelu vlastností se zobrazí právě barva výplně – obr.35.



Obr. 35: Změna barvy výplně

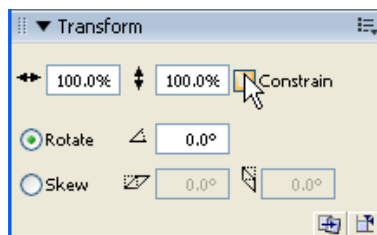
- Barva výplně i vlastnosti čar najednou: dvakrát klikneme na výplň – tím vybereme celý objekt.

- Vlastnosti jediné čáry: stačí na ni kliknout. Označí se právě tato jediná čára a ve vlastnostech budou zobrazeny její vlastnosti (viz kapitola Čára).
- Vlastnosti celého obrysu: pro změnu celého obrysu na něj musíme dvakrát kliknout.
- Jednoduchá změna tvaru (průhyb strany, natočení, deformace, změna velikosti, apod.):
 - Deformace strany – průhyb: postupujeme stejně jako u čáry – (opět kapitola Čára).
 - Natočení: dvojklikem označíme celý objekt, v tomto případě obdélník, a pak můžeme použít:
 - Nástroj Volná transformace (Free transform), najedeme-li na roh změní se kurzor (obr. 36) a můžeme točit dle libosti. Při zapnutém uchopování (Snap) se automaticky vyznačují polohy pootočení po 45°.



Obr. 36: Vzhled kurzoru při natočení

- Paletku Transformace, ve které můžeme přímo zadávat číselnou hodnotu natočení (obr. 37).
- Změna velikosti: při označeném objektu můžeme opět použít:
 - Nástroj Volná transformace: uchopíme za rohový klíčový bod a posouváním měníme velikost a poměr stran. Pokud chceme měnit pouze velikost při zachování poměrů, musíme držet při posouvání klávesu Shift.
 - Paletku transformace: opět zadáváme číselnou hodnotu zmenšení nebo zvětšení (procentuální). Požadujeme-li zachování poměru stran, musí být zapnuta volba Constrain – viz obr 37. V opačném případě lze měnit zvlášť poměr v ose X a Y.



Obr. 37: Paletka transformace

- Složité změny (změna počtu vrcholů, tvaru):
 - Změna polohy vrcholu: najedeme na vhodný kurzor (tvar kurzoru se změní – obr. 38), uchopíme jej přes LMB a přesuneme do nové polohy.



Obr. 38: Tvar kurzoru nad vrcholem



Obr. 39: Tvar kurzoru nad úsečkou

- Pro snížení počtu najedeme kurzorem na vhodný vrchol. Přetáhneme-li jej nad jiný vrchol a pustíme, dojde k jejich spojení.
- Chceme-li zvýšit počet vrcholů, musíme použít nástroje Volná transformace (tím dojde k vyznačení ohraničení) a pak nástroj Podvýběr – přes klávesu Ctrl přidáme další klíčové body, které pak můžeme posunout do požadované polohy. Pro větší počty vrcholů je však vhodnější použít nástroj Polygon (více v následující kapitole).

3.3 Polygon (Polygon)

U tohoto nástroje nejprve musíme nastavit, zda má být polygon vnější nebo vnitřní (tedy klasický n-úhelník nebo hvězda) a počet vrcholů. Poslední možnost - velikost počátku (Start point size) se týká pouze nastavení hvězda – jde o rozměr paty ramena (obr).



Obr. 40: Nástroj polygon (zleva): polygon; star(start point size) 0,2 a 0,7

Při kreslení se prvním kliknutím určí počáteční bod a tažením nastavujeme velikost polygonu a natočení. Se stisknutou klávesou Shift je zaručeno přesné nastavení.

3.4 Ovál (Oval)

Při kreslení se standardně začíná z rohu. Ano, i kulaté a zaoblené objekty mají ve Flashi rohy – je totiž nutné rozlišovat mezi obrysem (čára kolem výplně) a ohraničením (pravoúhlý obrazec, do kterého se vleze celý objekt).

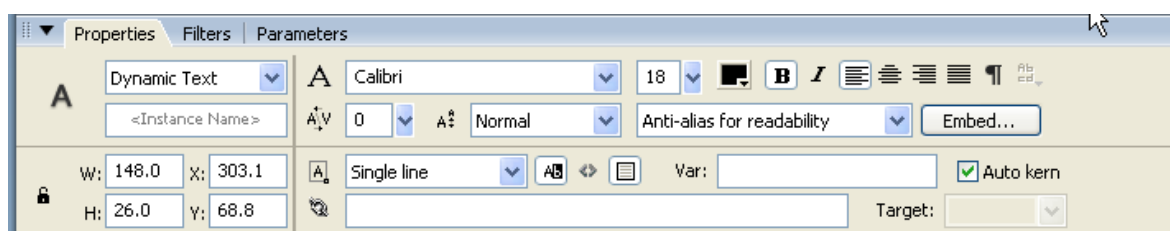
Potřebujeme-li z určitého důvodu kreslit od středu, použijeme naši známou klávesu Alt. S klávesou Shift tvoříme souměrné obrazce, v tomto případě kruhy.

Po prvním kliknutí, kdy určujeme první bod, pak tažením tedy určujeme velikost.

4 Texty

Bez textu se neobejde téměř žádná Flashová prezentace, proto jsou na něj kladeny vysoké nároky – při nevhodně použitém textu (typ písma, typ textového pole, ...) je nepřehledný, nečitelný atd..

Obecně platí: jakýkoliv objekt, jakákoliv křivka je popsána matematicky ve výsledném .swf souboru. Písmo, které využíváme při práci v programu je v podstatě totožné jako u standardních fontů. Má-li však být zajištěna 100% přenositelnost, musí být přeneseny i použité fonty. Toho lze dosáhnout jedině tím, že dotyčný font je „přibaleno“ do .swf (nemusí platit vždy - viz níže).



Obr. 41: Vlastnosti textu

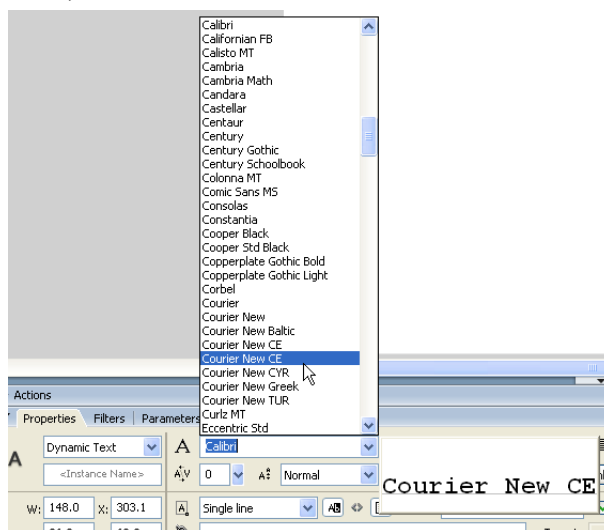
Při práci s textem si kromě typu pole (popsáno dále), vybíráme další atributy (obr. 41):

Typ textového pole – můžeme vybírat ze tří možností

- *statické textové pole* - jak už název napovídá, používáme jej pro text, který se nemá měnit – nadpisy, popisky tlačítek a podobně.

- *dynamické textové pole* – slouží k zobrazování textu, ale na rozdíl od statického pole, lze jeho obsah v průběhu prezentace měnit (pomocí AS)
- *vstupní textové pole* – stejně jako předchozí je plně interaktivní, slouží však k získávání textu od uživatele. Ten je poté zpracován pomocí AS. Vstupní pole je společně s dynamickým velmi často využíváno při tvorbě formulářů. Podmínkou pro plné využití však je pojmenování instance (popsáno v kapitole Symboly).

Typ písma (font) – neboli fonty, při výběru se automaticky zobrazuje vzorek písma, nad kterým přejíždíme (obr. 42).



Obr. 42: Výběr písma včetně okna vzorku

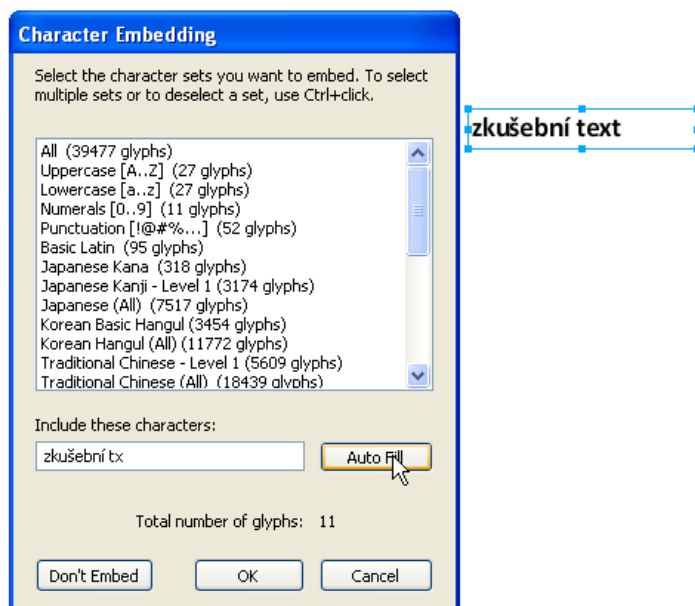
Máme na výběr ze dvou základních skupin:

- písma zařízení - nejde o font v klasickém pojetí. Při použití písma zařízení se při přehrávání najde v počítači uživatele nejpodobnější font a ten se využije. Písmo tedy nemusí být součástí .swf (nepatrné snížení velikosti souboru), ovšem autor nemá kontrolu nad výsledným zobrazením.

Flash používá tři základní písma zobrazení, jsou na seznamu fontů uvedena jako první a pro jasné rozlišení je prvním znakem názvu fontu _ (podtržítka) a na rozdíl od dalších fontů neobsahuje název velká písmena :

- `_sans` – odpovídá písmu Arial nebo Helvetica – **sans**
- `_serif` – odpovídá Times New Roman – **serif**
- `_typewriter` – odpovídá fontu Courier - **typewriter**

- vkládaná písma – lze vybírat ze všech fontů, které jsou nainstalovány v autorském PC. Toto písmo se pak vloží do .swf, takže sice zvýší velikost, ale zobrazení textu bude za všech okolností stejné. Pokud používáme vkládaná písma pouze v malé míře, můžeme velikost souboru snížit také tím, že do něj nenecháme vložit celou sadu (tj. všechny znaky, které font obsahuje), ale jen ty použité (obr 43) . Toto opatření má samozřejmě smysl pouze pro malé objemy textu – při delších textech se obvykle znaky opakují a je tedy použita další instance znaku z knihovny (viz kapitola Symbols).



Obr. 43: Vkládání vybraných znaků - dialog Embedding

Velikost písma (font size) – má stejný význam jako u běžných textových editorů. Lze ji měnit buď pomocí posuvníku v pravé části okýnka nebo přímo zadáním číselné hodnoty v bodech (pt). Velikost můžeme nastavit v rozsahu 8 až 96 pt.

Barva písma (text color) – po kliknutí se objeví klasická paletka barev, můžeme ale také využít nástroj Kapátko.

Formátování – v češtině někdy označováno stejně jako font – tj styl. Zde je Flash poněkud chudší, nabízí pouze možnost písma tučného (B – bold) nebo písma v kurzívě (I – italic).

Zarovnání – zarovnat text můžeme doleva, doprava, na střed a do bloku. Ve starších verzích Flashe (do MX 2004) bylo možno zarovnávat pouze celá pole, od verze 8 lze samostatně zarovnávat i jednotlivé řádky pole.

Nastavení formátování – určujeme odsazení řádku od okraje, vzdálenost mezi řádky, levý a pravý okraj (vzdálenost textu od okraje pole). Všechny hodnoty jsou udávány v bodech

(pt).

Orientace textu (text orientation) – opět můžeme volit z více možností (obr. 44):

- horizontální – klasický text beze jakýchkoliv úprav
- vertikální, zleva doprava – bez dalších úprav je písmo orientováno stále stejně, ovšem s většími rozestupy, takže je lze bez problémů natáčet
- vertikální, zprava doleva – pořadí znaků textu je zrcadlově otočené, jinak zachovává stejné vlastnosti jako předchozí nastavení (vertikální, zleva doprava)

t x e t í n b e š u k z zkušební text

zkušební text

Obr. 44: Různé orientace textu

U vertikálních nastavení se po volbě objeví v panelu vlastností další tlačítko – rotace (rotation). To nenabízí výběr z více možností, ale slouží spíše jako přepínač mezi orientací horizontální a vertikální. Při úpravách a práci s textem tedy přepínačem zapneme polohu horizontální a po dokončení pak znovu použijeme přepínač pro pootočení do polohy vertikální. Podle nastavení orientace textu se pak text pootočí buď o 90° doleva nebo doprava.

Vzdálenost písmen (letter spacing) – lze nastavit v rozsahu -60 až +60 bodů, pokud jej nebudeme měnit (tj. vzdálenost je nastavena na 0 bodů), použije se standardního nastavení – proporcionality písma. Určitá písmena jsou z estetických důvodů od sebe vzdálena více než jiná – např. OV, AV, IV. Požadujeme-li, aby vzdálenost všech písmen byla za všech okolností stejná, musíme ji ručně nastavit. Srovnajte na obr. 45

zkušební text zkušební text **zkušební text**

Obr. 45: Vzdálenosti písmen (zleva): běžná, zvýšená, snížená

Pozice písmen (charecter position) – volíme písmo normální, horní index, dolní index.

zkušební text **zkušební text** zkušební text

Obr. 46: Pozice písmen (zleva): horní index, normální, dolní index

Metoda zobrazení (font rendering method) – určujeme metodu, pomocí které má být daný text zobrazen – zda je určen pro zobrazení v animaci (anti-aliasing for animation), pro čtení (anti-aliasing for reading), jako bitmapa (no anti-aliasing), vlastní zobrazení (custom anti-aliasing) nebo použít zobrazení jako u fontů koncového zařízení (use devices fonts).

Podle seznamu možností je zřejmé, že fonty lze rozdělit na dvě základní skupiny – ty které využívají anti-aliasing (vyhlazování), a ty které jej nevyužívají.

Text s vyhlazením využívá jemných přechodů, stínování a dalších optických triků k tomu, aby měl uživatel dojem, že písmo je krásně zaoblené, bez ostrých hran (tato technologie bývá označována také jako TrueType nebo ClearType). Naopak text bez vyhlazení má ostré přechody, je hrbolatý – plně kopíruje neviditelnou pixelovou mřížku – viz obr. 47

Zde je napsán zkušební text Zde je napsán zkušební text

Obr. 47: Porovnání písma s antialiasingem (nahore) a bez něj

Výběr znaků (embeded characters) – toto nastavení se týká pouze dynamického a vstupního textového pole. Základ je popsán v odstavci Vkládaná písma. Zde už jen doplním, že při výběru znaků lze použít např. jen malá písmena, nebo jen čísla apod. Samozřejmě je možné kombinovat různá nastavení (velká písmena + interpunkce, ...). Další možností je využití funkce automatického doplnění (auto fill) – počítač projde celý text a použije jen ty znaky, které jsou opravdu použity.

Pro ilustraci použiji pro větu „Já mám rád koblihy, a ty?“ funkci automatického doplnění. Výsledkem bude řetězec použitých znaků „já mrdkoblihy,at?“ - každý použitý znak je použit pouze jednou a uložen do knihovny, ze které se při dalším výskytu pouze zkopíruje.

Odkazy (URL link) – do této kolonky vpisujeme adresy případných dotazů. Pro emailovou adresu zadáváme s prefixem mailto: např. <mailto:som@seznam.cz>.

Pro odkaz na internetovou stránku zapíšeme prostě tuto adresu, ovšem vždy musí být i s popisem protokolu – tedy <http://www.seznam.cz>. Pokud by adresa neobsahovala řetězec <http://>, odkaz by nefungoval!

Vyrovnání dvojic znaků (kerning) – jedná se o rozestupy mezi určitými znaky – OV, AV. Bližší popsáno v odstavci Vzdálenost písmen.

Možnost výběru (selectable) – opět pouze pro dynamický a vstupní text. Při použití této volby lze text označovat kurzorem, např. pro vykopírování do jiné aplikace.

Cíl (target) – zde nastavujeme, jak se zobrazí případný odkaz – v novém okně, v rodičovském okně atp.

Možnosti nastavení textu jsme ani zdaleka nevyčerpali, pro naše účely však jako úvod do problematiky určitě postačují.

5 Symboly

Cokoliv ve Flashi vytvoříme, jakýkoliv objekt (grafický obrazec, animace, text, ...) můžeme využít jako symbol (buď jej vytvoříme v režimu symbol nebo jej na symbol převedeme kdykoliv později). Vytvořené symboly jsou ukládány v knihovně symbolů, ze které je můžeme znovu a znovu opakovaně využívat, editovat, mazat a vůbec provádět jejich management (více o knihovnách v samostatné kapitole). Z knihovny symbolů, pak autor „vytahuje“ jen jakési odkazy – tzv. instance.

Pokud si jednotlivé instance pojmenujeme pomocí panelu vlastností, můžeme je ovládat pomocí AS nebo chování (behaviors).

Tento postup má velký význam tam, kde používáme prvky, které se často opakují, kde vyžadujeme jednotný vzhled (např. navigační tlačítka u rozsáhlejší prezentace). Nejenže ušetříme čas tím, že symbol vytvoříme pouze jednou a pak na něj odkazujeme, takže nemusíme opakovaně tvořit totéž, ale také můžeme podstatně redukovat velikost výsledného souboru. Při jakémkoliv požadavku na změnu navigace pak stačí změnit pouze symbol uložený v knihovně a tím změníme i všechny instance, takže odpadá časově náročné procházení celé prezentace a hledání „sirotků“. Další nespornou výhodou je snížení velikosti .swf souboru – symbol je uložen pouze jednou (popsáno v předchozích kapitolách).

Některé učebnice a výukové materiály doporučují na symboly převádět veškeré objekty použité v prezentaci – knihovna prvků je totiž při přehrávání načtena do paměti celá, takže při požadavku přehrávače na zobrazení určitého objektu je tento již načten v paměti a odpadá časová prodleva pro načtení z disku nebo jiného média (obvykle podstatně pomalejšího než operační paměť PC).

Za určitých podmínek lze dokonce zajistit, aby se jednotlivé instance symbolu od sebe lišily – můžeme přidávat různé efekty (barevné filtry, změna průhlednosti, ...), pomocí

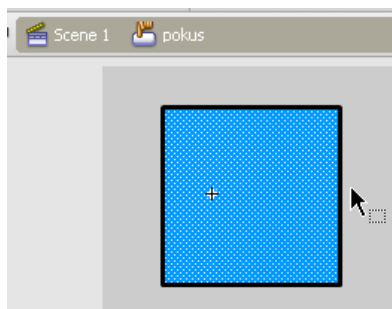
volné transformace je deformovat atd. S úspěchem tedy lze symboly použít, aniž by jejich opakování bylo do očí a vytvářelo dojem nudné prezentace a „odfláknuté“ práce.

Symboly lze vytvářet z existujících objektů pomocí klávesové zkratky F8 – nejprve je samozřejmě musíme označit, pak použijeme klávesu F8 (lze použít i příkaz z nabídky Insert/Convert to symbol). V následujícím dialogu musíme vybrat typ symbolu a registraci (za okamžik vysvětlím), potvrdíme a vše je hotovo.

Pokud převádíme na symbol např. čtverec (víme že má obrys a výplň), vznikne z primitivní grafiky (zvlášť obrys, zvlášť výplň) na pracovní ploše jediný objekt – jako bychom obrys a výplň sdružili do jediné skupiny.

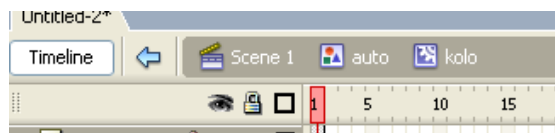
Chceme-li změnit již hotový symbol, máme opět několik možností:

- V knihovně symbolů – dvojklikem nebo vyvoláním nabídky přes RMB. Při tomto způsobu se na pracovní ploše otevře scéna upravovaného symbolu a my nevidíme žádné další objekty ani plochu hlavní scény
- Na místě – dvojklikem poklepeme na instanci měněného symbolu, tím se dostaneme do režimu editace symbolu, který je zvýrazněn. Ostatní prvky zešednou a jsou neaktivní. Tento způsob je vhodný tam, kde potřebujeme dosáhnout přesného umístění symbolu vůči jiným prvkům – viz obr.



Obr. 48: Editace symbolu na místě

Oba tyto způsoby, resp. režimy úpravy symbolu jsou indikovány v nejspodnější části panelu Časová osa. První popiska (v našem případě Scene 1) určuje, ve které scéně se právě nacházíme. Další popisky pak naznačují, v jakých symbolech se pohybujeme. Pořadí popisek určuje vztahy mezi jednotlivými symboly – viz obr. 49.



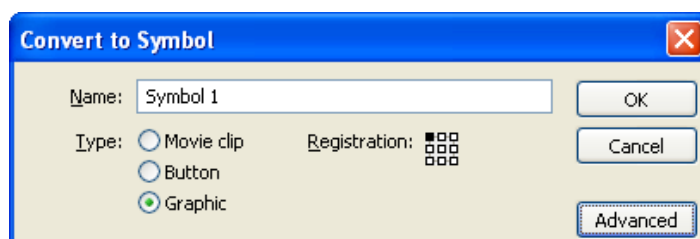
Obr. 49: Orientace ve vnořených symbolech

Při jakékoliv změně symbolu (respektive instance symbolu) je nutné si uvědomit, že jakákoliv změna se neprojeví jen u dotyčné instance, ale u všech instancí symbolu! Pokud potřebujeme provést změnu, která postihne pouze jednu jedinou instanci, můžeme použít pouze transformaci a efekty – více v kapitole Úpravy instancí symbolů, Efekty. Nutno podotknout, že tyto změny jsou spíše jednodušší.

Typy symbolů

Existují tři kategorie symbolů (podrobně je popíší v dalších kapitolách):

- Klip (Movie clip)
- Tlačítko (Button)
- Grafika (Graphic)



Obr. 50: Dialog konverze na symbol (základní možnosti)

Typ symbolu lze kdykoliv později změnit, záleží však na aktuálním způsobu používání symbolu (může nastat problém např. při převodu z tlačítka na grafiku, pokud již využíváme funkce tlačítka). Každá kategorie má specifické vlastnosti, bez jejichž znalosti by nešlo tvořit kvalitní prezentace.

Pro všechny symboly je ale společná možnost využívání vrstev a jejich složek, mohou obsahovat další (vnořené) symboly – samozřejmě kromě sebe sama.

Všechny symboly mají také vlastní panel vlastností, který je v případě nutnosti popsán u jednotlivých typů.

Panel má však vždy tlačítko SWAP – jedná se o nástroj, pomocí něhož můžeme stávající instanci symbolu vyměnit za jakoukoliv jinou (ve smyslu záměny za jiný zdroj instance z

knihovny). Instance se pak sice změní, ale zůstane na původním místě se všemi svými nastaveními (změní se pouze obsah).

Registrace

Výstižnější název by zřejmě byl umístění symbolu nebo zarovnání symbolu. Pomocí jednoduché matice bodů (3x3) – viz obr. 50 - můžeme kliknutím určit zarovnání kresby uvnitř symbolu.

5.1 Klip (Movie Clip)

Klip, dále jen MC, je zřejmě nejpoužívanější. Jako jediný má nezávislou časovou osu – takže animace se přehrává bez ohledu na hlavní časovou osu. Tzn. mimo jiné, že můžeme vytvořit opakující se (cyklickou) animaci nebo vybudovat aplikaci ve které je její kompletní obsah umístěn kompletně v MC. Takový klip pak může být přehrán na ploše přesto, že hlavní časová osa obsahuje pouze jediný snímek.

Typickým příkladem MC a v něm vnořených dalších MS je jedoucí auto:

- nejprve je vytvořen jednoduchý MC (nazveme jej např. *kolo_mc*), ve kterém se otáčí kolo osobního automobilu
- pak vytvoříme nový klip *auto_mc*, ve kterém bude nakreslena kapota vozu a přidáme MC otáčejícího se kola
- pokud v tomto bodě spustíme přehrávání (Ctrl + Enter), uvidíme, že auto sice stojí na místě, ale už se mu otáčejí kola
- nyní spojíme karoserii a obě kola do jediné skupiny (Ctrl+G), tím zajistíme, že kola se vůči autu nepohnou
- v dalším kroku vytvoříme v *auto_mc* animaci, ve které se celá skupina přesune z jednoho místa do druhého.

Vznikla tak animace, ve které se pohybují tři objekty – karoserie auta a dvě kola, která ještě navíc rotují kolem vlastní osy.

5.2 Tlačítko (Button)

Zkratka BTN. Tento symbol obsahuje vždy pouze 4 snímky (počet vrstev však není omezen), pomocí kterých určujeme vzhled chování tlačítka v součinnosti s kurzorem.

Po vytvoření nového tlačítka jsou všechny snímky prázdné a je potřeba vytvořit alespoň jeden (a přidat kód pro nastavení chování), aby tlačítko začalo plnit svou funkci.

Popis snímků tlačítka (jak již bylo řečeno, jsou vždy právě 4 a jejich pořadí se nemění – obr. 51):

UP

První snímek, obsahuje podobu tlačítka v klidu (není zmáčknuto ani se nad ním nenachází kurzor).

OVER

Druhý snímek, uchovává podobu tlačítka při najetí kurzorem nad tlačítko (aktivace).

DOWN

Třetí snímek, zobrazuje podobu tlačítka při stisknutí tlačítka myši.

HIT

Poslední, čtvrtý snímek. V něm určujeme aktivní oblast – tzn. oblast, která reaguje na najetí kurzoru nebo zmáčknutí tlačítka.

Tato oblast je při běžící animaci neviditelná, ale právě ona je nejdůležitější a postačuje ke zprovoznění tlačítka. Pokud však zůstane tento snímek neobsazen, je aktivní plocha automaticky vygenerována z prvního snímku (UP).



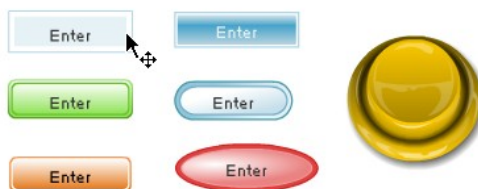
Obr. 51: Klíčové snímky symbolu Tlačítko

Snímky mohou obsahovat libovolné prvky – od primitivní grafiky (barevný obdélník) až po složité MC, zvuky atd.

Kromě vlastních tlačítek lze využít i univerzální tlačítka připravená tvůrci programu. Ta se nacházejí v knihovně tlačítek (nabídka Window/Common Libraries/Buttons), kde jsou tematicky rozdělena do několika kategorií – pro ilustraci uvádím několik takových tlačítek na obr. 52.

Tlačítko lze využít ve dvou režimech:

- Track as Button – chová se jako normální tlačítko
- Track as Menu – tlačítko využíváme jako přístupový bod k nabídkovému menu



Obr. 52: Ukázky předhotovených tlačítek

5.3 Grafika (Graphic)

Zkratka GPH, je zřejmě nejjednodušší symbol. Přestože v něm můžeme tvořit animace, jeho časová osa je závislá na hlavní časové ose – pokud tedy v GPH vytvoříme animaci trvající 50 snímků a umístíme ji do hlavní scény, která má pouze 45 snímků, naše animace uvnitř GPH se nepřehraje celá.

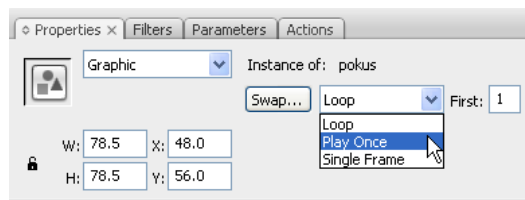
GPH je tedy vhodná k uchování statických objektů – pozadí scén, nepohyblivé objekty apod.

Pokud máme vytvořen MC a chceme jej z nějakého důvodu použít jako statický symbol, změníme její typ na GPH. V panelu vlastností pak máme 3 možnosti přehrávání (obr. 53):

Loop – animace bude přehrávána jako smyčka i v případě, že je ukončena příkazem *stop*. V okýnku First můžeme nastavit první snímek, od kterého se začne animace přehrávat

Play once – animace je přehrána pouze jednou, a to od snímku uvedeném v okýnku First.

Single Frame – bude zobrazen právě jeden jediný snímek – ten, který je uveden v okýnku First.



Obr. 53: Možnosti přehrávání

5.4 Úpravy instancí symbolů, Efekty

Plnohodnotná úprava instance symbolu není už z principu možná (jakákoliv větší změna se okamžitě projeví u všech instancí). Máme tedy jen omezené možnosti: smíme použít pouze možnosti panelu Transformace nebo nástroje Efekty ve vlastnostech.

Transformace

stejně jako u běžných objektů lze měnit velikost (s nebo bez zachování poměru stran), natočení, zkosení instance. Tyto funkce byly již vysvětleny v kapitole Nástroje.

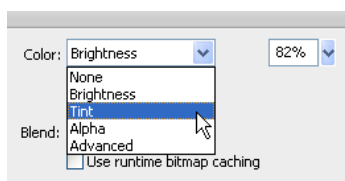
Při úpravách instance symbolu však navíc můžeme využít další nástroje – dvě nová tlačítka nalezneme v pravém spodním rohu paletky:

Kopíruj a použij (Copy and Apply Transform) – po nastavení všech požadovaných úprav a stisknutí tohoto tlačítka se na ploše vytvoří nová instance původního symbolu, která má všechny nově nastavené vlastnosti. Původní instance zůstává samozřejmě nedotčena na svém místě.

Obnovení původního nastavení (Reset) – při použití u transformovaných instancí obnoví původní nastavení.

Efekty

Efekty nastavujeme v panelu vlastností – viz obr. 54. Nelze však použít více efektů najednou – vždy může být aktivní pouze jeden. Standardně jsou efekty samozřejmě vypnuty (Effect None).



Obr. 54: Nastavení efektů

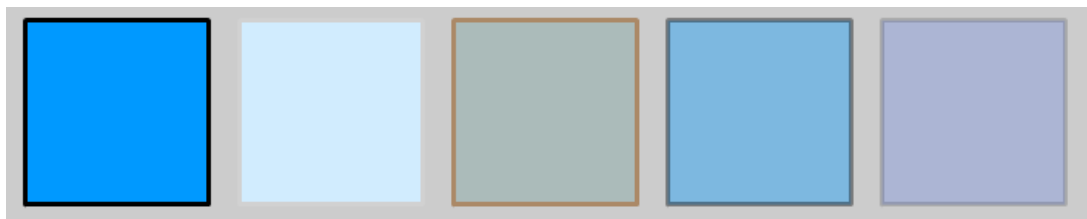
Můžeme však využívat následující:

Jas (Brightness) – zvyšujeme/snižujeme jas celé instance symbolu v rozsahu -100 až +100

Zabarvení (Tint) – tento efekt si lze představit, jako bychom přes instanci položili barevnou folii, takže instance má jinou barvu nebo odstín. U tohoto efektu nastavujeme barvu a sytost „fólie“.

Průhlednost (Alpha) – pracuje jako Alpha u barevných výplní, ovšem pozor – zde měníme průhlednost CELÉHO objektu (kromě výplní i čar a obrysů).

Pokročilá nastavení (Advanced properties) – v podstatě jde o kombinaci všech předchozích efektů – v levé části nastavujeme barvu, do které se má objekt přebarvit a průhlednost. V pravé části nastavujeme jas jednotlivých barevných složek (R, G, B) a celkovou jasnost.

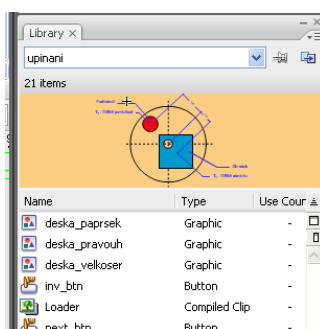


Obr. 55: Efekty symbol, zleva: původní symbol, pak efekty jas, zabarvení, průhlednost, pokročilé nastavení

6 Knihovny

6.1 Význam, funkce

Každý existující symbol je uložen v knihovně symbolů, ze které jej můžeme opakovaně vkládat do projektu. Knihovna je tvořena automaticky a má dvě části: náhled a správce – obr. 56.



Obr. 56: Knihovna

Objekt v knihovně však nutně nemusí být použit – knihovna může být použita i jako zásobník objektů a symbolů ([15]).

V pravém horním rohu se nachází kontextové menu, pomocí něhož pracujeme s položkami knihovny. Nabídka obsahuje následující položky:

Nový symbol (New symbol) – vyvolá nám již známý dialog pro tvorbu nového symbolu

Nová složka (New folder) – vytvoří nový adresář, do kterého pak můžeme přetahovat jednotlivé symboly, takže pomáhá udržovat přehled

Nový font (New font) – umožňuje definovat font jako symbol a pak jej využívat ve sdílených knihovnách

Nové video (New video) – vytvoří nový video soubor, do kterého pak můžeme importem vložit videosekvenci

Přejmenovat (Rename) – přejmenuje vybraný symbol (a odkazy v použitých instancích)

Přesun do nové složky (Move to new folder) – bude vytvořena nová složka a symbol je do ní přesunut

Odstranění (Delete) – vymaže symbol

Editace (Edit) – otevře samostatné okno symbolu pro editaci

Editace v ... (Edit with) - symbol lze editovat v programech třetích stran

Vlastnosti (Properties) – otevře panel podobný panelu při tvorbě symbolu, v něm lze měnit další vlastnosti

Sdílení (Linkage) – slouží ke sdílení knihovny pro další prvky.

Kromě těchto základních existují i další (aktualizace, přehrát, otevřít/zavřít složky atd.). Jejich výčet však přesahuje rámeček této práce.

6.2 Sdílení knihoven

Pro snazší práci s více soubory lze použít i technologii sdílení knihoven – v rozsáhlém projektu jako je tento jde doslova o dar z nebe. Pokud chceme tuto možnost využívat, musíme:

- Existující symbol „nalinkovat“ - v knihovně vyvoláme přes RMB nabídku, ze které vybereme položku sdílení (linkage), v dialogu pak potvrdíme sdílení a uvedeme název symbolu, pod kterým jej budeme sdílet. Nakonec zadáme cestu k souboru, ve kterém je symbol sdílen.
- Jelikož lze knihovny sdílet jen ze souboru .swf, musíme jej vygenerovat
- Když pak otevřeme nový dokument, musíme nejprve otevřít sdílenou knihovnu (File/Open as shared library).

Tento způsob nejen že snižuje velikost výsledné prezentace (symbol je v ní uložen jen jednou)

Kromě tohoto postupu lze také v knihovně jen přepnout mezi knihovnami otevřených souborů a symbol prostě vložit. Takto přetažený symbol je ale pouze překopírován a v nové knihovně znova uložen. Pokud navíc potřebujeme provést změnu symbolu, musíme ji provést ve všech knihovnách zvlášť.

6.3 Import

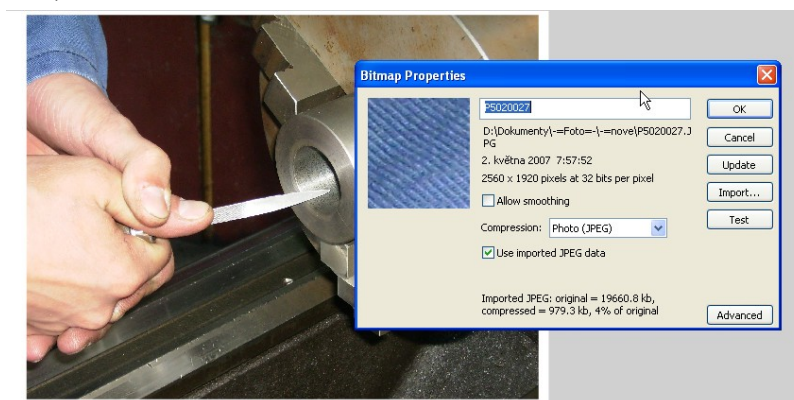
tato funkce značně rozšiřuje možnosti programu – nejsme odkázáni jen na výtvary z Flashe, ale můžeme využít i složitější podklady.

Soubory (video, zvuky, obrázky, ...) lze vkládat přímo na plochu (Import to stage) nebo do knihovny symbolů (insert to library) a použít je později.

Import bitmap

Využívá se především pro nahrání obrázků, které pak využíváme jako podklady scén apod. Jde především o formáty BMP, GIF, JPG, JPEG, PNG, ale i další méně známé.

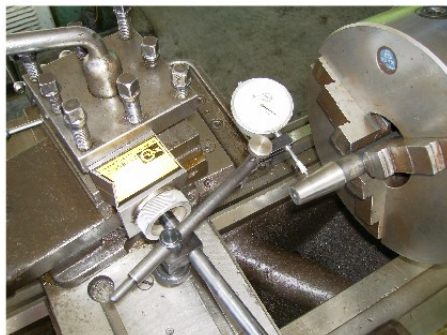
Bitmapa je po importu automaticky komprimována do formátu JPEG nebo GIF. Pokud si vyvoláváme vlastnosti bitmapy z knihovny, můžeme mimo jiné měnit nastavení této komprese (obr. 57).



Obr. 57: Nastavení komprese bitmapového obrázku (musí být v knihovně symbolů)

Pokud pak navíc provedeme konverzi bitmapy na symbol, vzniká ohraničený objekt, který je vyplněn bitmapovou výplní (lze měnit ohraničení a výplň se automaticky přizpůsobí).

Velice zajímavých efektů můžeme dosáhnout převodem bitmapy na vektory (tzv. trace). Dojde tím k vytvoření vektorových ohraničení s původní výplní. Tento postup se příliš nehodí k úpravě fotografií – obr .



Obr. 58: Bitmapa - bez úprav



Obr. 59: Bitmapa - trace 3

Import vektorů

Import vektorů je zřejmě nejlepší volba – objekt se automaticky neukládá do knihovny, ale je plně editovatelný pomocí standardních nástrojů obsažených ve Flashi. Navíc je jeho velikost nesrovnatelně menší.

Mezi nejznámější vektorové soubory patří především FreeHand (nejkvalitnější import), PNG, AI a EPS (Adobe Illustrator), DXF (AutoCAD) a mnoho dalších.

Import zvuku

Při importu je zvuk vždy umístěn pouze do knihovny, ze které si jej musíme vyvolat. Importovat lze pouze soubory WAV, AIFF, MP3. Nabídka je tedy podstatně chudší než pro import grafiky.

Importovaný zvuk je stejně jako bitmapa komprimován (obvykle do MP3) a stejně jako u ní, můžeme i u zvuku nastavovat kvalitu komprese – v knihovně vyvoláme vlastnosti zvuku přes RMB.

Import videa

Pro import je využíván kodek Sorenson SPARK (také označován jako H.264 - [18]) - jednoduchá verze je zabudována přímo do Flashe, lze však zakoupit i verzi PRO, která je dobře uzpůsobena pro integraci).

Při importu videa nastavujeme především kodek, kvalitu, četnost klíčových snímků (keyframes). Video lze také v omezené míře stříhat a vytvářet i efekty.

Import je možný pro formáty MOV, AVI, MPG, MPEG, ASF.

7 Animace

Animování je tvorba iluze pohybu. Stejně jako u filmu tedy nevytváříme pohyb v pravém slova smyslu, ale pouze statické stavy objektů, které jsou však natolik blízko u sebe a přehrávány tak rychle, že lidské oko je vnímá jako pohyb.

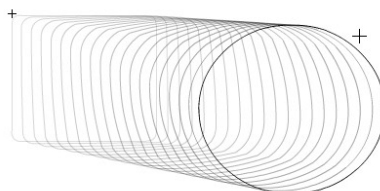
Animovat můžeme několika způsoby: políčko po políčku, motion tween, shape tween.

7.1 Animace políčko po políčku

Nejpracnější, ale zároveň nejpřesnější způsob tvorby animací, kdy v každém políčku posuneme objekt o malý kousek vedle.

7.2 Animace pomocí Tween shape

Patří do skupiny tzv. tweenů (existuje ještě tween motion) – pomocí těchto nástrojů si můžeme ulehčit život s animacemi – tweeny automaticky přepočítávají přechody mezi počáteční a koncovou polohou objektu.



Obr. 60: Shape tween se změnou polohy (z obdélníku do kružnice)

Shape tween (dále jen TS) je zaměřený na změny tvaru obrysů – jinými slovy, používáme jej k vytvoření plynulého přechodu z jednoho tvaru objektu do jiného (obr 60), případně změna barvy apod.

Při práci vytvoříme dva klíčové snímky na časové ose. Jejich vzdálenost závisí na požadované délce animace. Po kliknutí na první klíčový snímek se v panelu vlastností objeví nabídka Tween: none. Pokud zvolíme shape, snímky zezelenají a objeví se přes ně šipka – znamení TS (obr 61).



Obr. 61: Indikace Shape tween

TS si sám vytvoří „optimální“ postup transformace (efekt je velmi podobný morphingu). Záměrně píší optimální v uvozovkách, protože výsledky nemusí být vždy optimální (respektive jsou optimální pouze u velmi jednoduchých objektů).

Pomocné body (shape hints)

Pokud chceme provést transformaci podle vlastních představ, je často potřeba použít pomocné body (shape hints). Jednoduše řečeno jde o dvojice značek, jednu umístíme na počáteční tvar a druhou na odpovídající bod konečného tvaru. Program pak pochopí (konečně), jak že to chtěl autor transformovat. Bohužel jediný bod stačí málokdy, takže dvojic je potřeba použít více – čím víc, tím líp.

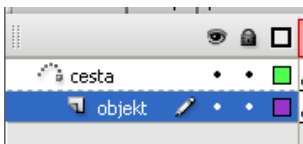
7.3 Animace pomocí Motion tween

Jak už název napovídá je Motion tween (dále jen MT) vhodný ke změnám polohy – tedy pohybové animace. Postupujeme podobně jako u ST – vytvoříme počáteční a koncový stav, tedy krajní polohy těles. Na snímky mezi nimi pak aplikuje MT. Klikneme na první klíčový snímek a z nabídky v panelu vlastností vybereme tween: motion. Snímky změni barvu (budou fialové) a opět se objeví šipka (obr 62).



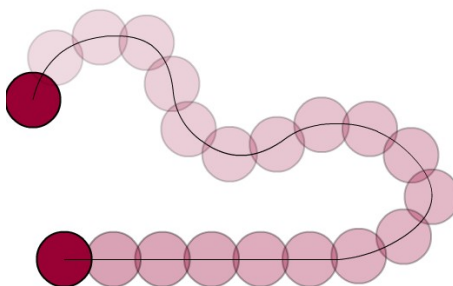
Obr. 62: Indikace Motion tweenu

Specialitou MT je pohyb po křivce. Zde potřebujeme vrstvu, ve které jsou umístěny krajní polohy symbolů, ale navíc i vrstvu, ve které je uložena vodící křivka. Máme-li obě, můžeme vrstvu s křivkou označit jako vodící (guide) a z vrstvy pod ní se automaticky stává vrstva vedená (obr. 63).



Obr. 63: Vrstva vedoucí
a vedená

Pak ještě musíme objekty přichytit ke koncům křivek. Po nastavení MT už zbývá jen označit políčko Orientovat k cestě (orient to path) - a animace je hotová (obr. 64)



Obr. 64: Motion tween po křivce

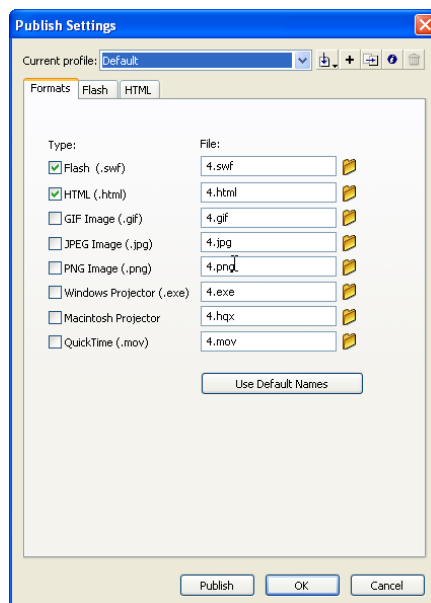
8 Publishing

Pokud je projekt až v této fázi, může si autor blahopřát – prezentace je hotová a zbývá ji jen „vypustit do světa“.

Nativní typ souborů, tzn .fla jako zdrojový kód není právě nejvhodnější pro uvolnění – jednak je relativně velký, ale hlavně není optimalizovaný. Proto musí vytvořit novou verzi, která je sice založena na .fla, ale zároveň je vhodná pro přehrání.

Možností exportu je více, ale každý mi asi dá za pravdu, že pro publikování Flashe má největší význam soubor .swf nebo .html.

Vlastnosti publikačních souborů nastavujeme pomocí dialogového okna Nastavení publikace (Publish settings), které najdeme v nabídce File/Publish settings – obr .



Obr. 65: Nastavení publikace

Na první záložce vybíráme, do jakých souborů chceme publikovat a na dalších záložkách (každý vybraný typ publikačního souboru má svou záložku) pak zadáváme konkrétní nastavení pro ten který soubor.

8.1 Export do .swf

Soubor .swf je vůbec nejlepší volba – jelikož je to nativní soubor Flashe, funguje v něm všechno jak se sluší a patří.

Záložka nastavení Flash má několik možností nastavení:

Verze (version) – určujeme, pro jakou verzi přehrávače má být prezentace určena. Obecně platí, že čím nižší verze přehrávače, tím vyšší kompatibilita, na druhou stranu ale nemusí ve starší verzi fungovat nejnovější technologie.

Pořadí nahrávání (load order) – pořadí, v jakém budou nahrávány jednotlivé vrstvy dokumentu – buď shora dolů, nebo naopak. Volba závisí na systému organizace vrstev.

Informace o velikosti (generate size report) – při zaškrtnuté volbě bude automaticky vygenerován .txt soubor s kompletními údaji o exportovaném souboru (velikost, počet snímků, doba, fonty, symboly,...)

Ochrana proti importu (protect from import) – volíme, pokud nechceme aby šel náš produkt naimportovat do jiného dokumentu (importér by tak získal naše symboly, postupy, zkratka téměř vše jako v původním .fla)

Kromě těchto lze ještě nastavovat obecné komprese videa, zvuku i obrázků, vzdálené ladění skriptů, aj.

8.2 Export do .html

Při exportu do html sice vytvoříme stránku s HTML kódem, nicméně po jeho prozkoumání zjistíme, že html vytváří jen jakousi obálku kolem našeho starého známého .swf souboru (řádek č. 12 ... src=pokus.swf ...).

```
1 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-
  8859-1" />
4 <title>pokus</title>
5 </head>
6 <body bgcolor="#ffffff">
7 <!--url's used in the movie-->
8 <!--text used in the movie-->
9 <!-- saved from url=(0013)about:internet -->
10<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
  codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash
  /swflash.cab#version=8,0,0,0" width="550" height="400" id="pokus"
  align="middle">
11<param name="allowScriptAccess" value="sameDomain" />
12<param name="movie" value="pokus.swf" /><param name="quality"
  value="high" /><param name="bgcolor" value="#ffffff" /><embed
  src="pokus.swf" quality="high" bgcolor="#ffffff" width="550"
  height="400" name="pokus" align="middle"
  allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
  pluginspage="http://www.macromedia.com/go/getflashplayer" />
13</object>
14</body>
15</html>
```

V záložce HTML můžeme volit (kromě jiného) způsob, jakým bude .swf vložen do html kódu, velikost animace po načtení, možnosti přehrávání, kvalitu zobrazení atd.

9 Action script

AS je samostatný programovací jazyk [19], určený přímo pro Flash. Pomocí něj můžeme projektům přidávat interaktivitu – použít ovládací prvky, formuláře...

AS vychází z Java Scriptu, který má prapůvod v jazyku C [15]. As je tedy objektově orientovaný jazyk, na rozdíl od pokročilejších verzí však nepotřebuje kompilátor – pro svůj běh vyžaduje engine, který vykonává příkazy. Může to být přehrávač (Flash player),

internetový prohlížeč s pluginem, ...

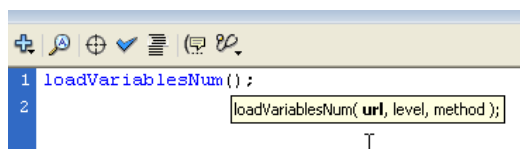
Engine vykonává příkazy postupně, jak přicházejí do fronty – takovým jazykům říkáme skriptovací.

Přesto, že AS vychází z Java Scriptu (dále jen JS), v několika podstatných vlastnostech se liší – nepodporuje tolik objektů jako JS, nepatrně se liší syntaxe konstruktorů, jinak pracuje s nedefinovanou funkcí, nedefinovanými číselnými proměnnými atp.

Příkazy lze přiřadit jednotlivým objektům na scéně (především tlačítkům), ale můžeme je zapsát i do klíčových snímků – záleží na použití skriptu. V případě zápisu do klíčových snímků je velmi nutné vytvořit pro AS samostatnou vrstvu – kód je tak shromážděn v jednom místě a odpadá jeho zdlouhavé vyhledávání (i když i na to existuje nástroj).

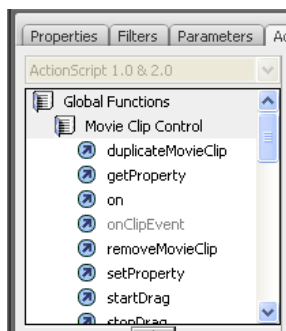
Programovat můžeme několika způsoby:

Ruční psaní (tzv expert mod) – v panelu actions. Tento způsob však vyžaduje hlubší znalosti AS – především příkazů. Při psaní se zobrazuje dynamická nápověda, především syntaxe.



Obr. 66: As - Expert mod

Vybírání příkazů (tzv. normal mod) – obr. 67 – vedle panelu actions je panel se strukturovaným seznamem příkazů – v horní části si vybereme, jakou verzi AS chceme použít, poté si v seznamu najdeme požadovaný příkaz a po dvojkliku dojde k jeho vepsání do okna editoru. Na uživatele tak zbývá už jen doplnit atributy příkazu.



Obr. 67: AS - Normal
mod

Script asistent – jde o předchozí způsob (Vybírání příkazů) jen s tím rozdílem, že asistent zároveň nabízí v dialogovém okně seznam možných atributů příkazu.

Chování (behaviors)– není programováním v pravém slova smyslu – zjednodušeně řečeno, pro každý prvek existuje omezené množství předprogramovaných příkazů, pomocí kterých nastavujeme chování tohoto prvku. Práce je podobná jako při využití Script asistenta.

Jelikož není AS v této práci příliš využit (zatím), nebudu se jím dále zabývat. O chování se ještě zmíním v kapitole Action Script, chování v praktické části.

II. PRAKTICKÁ ČÁST

TVORBA ENCYKLOPEDIIE

Při tvorbě samotné bylo potřeba nejprve určit nároky, které má práce splňovat. Ty byly celkem jasné: vytvořit prezentaci, která by měla splňovat hlavně pedagogická programátorská kritéria jako:

- je určena pro žáky středních škol
- obsahuje pro studenty podstatné informace o Technologii soustružení
- nemá vysoké nároky na HW ani SW
- musí mít co nejjednodušší ovládání
- pro větší přínos by měla obsahovat spíše obrázky, animace a video – zkrátka musí studenty zaujmout (neměla by obsahovat pouze text)
- je vypracována pomocí technologie Flash
- po dokončení 1. fáze musí být bez problémů šiřitelná
- velikost by neměla přesáhnout 128MB (minimální kapacita dnes prodávaných USB flash disků)
- v kterékoliv fázi vývoje je dále rozšiřitelná (funkce vyhledávání, testy, databáze výsledků, propojení přes webové služby, ...)

Po zhodnocení těchto nároků nastala další fáze – tvorba designu. Ten se týká také navigace – to byl zřejmě největší problém.

Obecně se dá říct, že na této práci byly nejtěžší dvě věci: nejprve vymyslet takovou strukturu, aby do ní bylo možno kdykoliv vstupovat a doplňovat, v ideálním případě měnit celé bloky obsahu (nejen informační, ale i programový). Druhým oříškem bylo vytvoření vhodného designu – aby byl jednoduchý, nerušil a přitom splnil všechny nároky na něj kladené.

Naplnění informačního obsahu již bylo spíš otázkou časové náročnosti a trpělivosti než čehokoliv jiného.

1 Návrh řešení

Stejně jako většina autorů, kteří se poprvé věnují rozsáhlejšímu projektu, jsem si myslel, že celý projekt udělám kompletně z „jedné vody“. Představoval jsem si, jak bude můj projekt obsáhlý, které technologie použiji a že po dokončení projektu se bez mé encyklopedie

žádný student neobejde.

Mylnost tohoto názoru se naštěstí ukázala velmi brzy, a já se opět dotkl pevné země. V tomto okamžiku začalo plánování, které (jak se později ukázalo) mělo opravdu klíčový význam pro další práci – jak už se to psává v knihách.

Prvním oříškem bylo vhodné zkombinování designu, programování a ovládání tak, aby design nerušil, ovládání bylo jednoduché a intuitivní a hlavně programování bylo co nejjednodušší :).

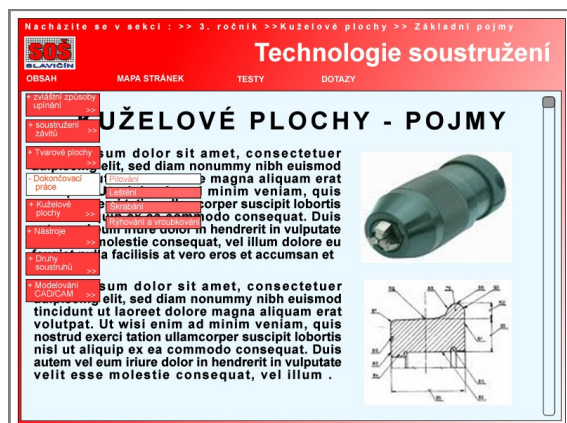
První verze byla tedy opravdu jednoduchá – obr. 65. Ústředním bodem celé prezentace je v horní části banner, z velké části zaplněn fotografií ovládacího panelu CNC (jde o nejnovější přírůstek dílenského vybavení naší školy) – tato fotografie se objevuje také v záhlaví nových webových stránek školy. Fotografií je pět a mají charakterizovat naši školu. Ve spodní části banneru se pak nacházejí navigační tlačítka.



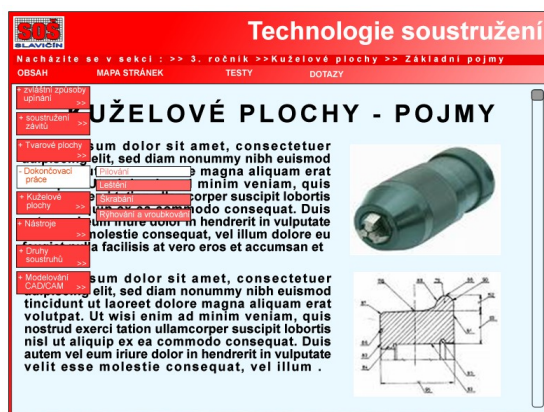
Obr. 68: Grafický návrh č. 1

Po dalších úvahách a konzultaci s kamarádem grafikem jsem však tuto variantu zavrhl jako příliš strohou.

Druhá verze byla kompletně přepracována, objevuje se roletkové dynamické menu, z hlavičky mizí fotografie – obr. 69. Bohužel orientační proužek v horní části byl příliš nevýrazný, takže přišla **třetí verze** (obr. 70)– proti předchozí došlo ke změně polohy orientačního proužku a jeho zvýraznění.



Obr. 69: Grafický návrh č.2



Obr. 70: Grafický návrh č.3

Přes počáteční nadšení se bohužel ukázalo, že ani tato verze není ta pravá – roletkovým menu, které je zcela určitě efektivní, se paradoxně zhoršuje orientace – nejen že uživatel musí rozevírat více úrovní, ale ještě navíc musí sledovat další informace v orientačním řádku, aby vůbec věděl, ve které části prezentace se nachází.

Dalším důvodem odmítnutí byl posuvník na pravé straně textové části – po úvaze jsem si uvědomil, že pokud se bude danému tématu věnovat pouze jedna stránka, bude značně dlouhá a tím pádem nepřehledná. Zhorší se nejen přehlednost, ale i zrakové vyhledávání požadovaných informací.

Takže bylo rozhodnuto, že prezentace musí mít stránky tak velké (respektive tak malé), aby byl celý její obsah vidět okamžitě, podobně jako je tomu u různých prezentačních nástrojů (Powerpoint, Oo Impress, ...). Tím se ale rapidně zvýšil počet stránek, čemuž roletové menu naprosto nevyhovuje – prudce by se zvýšil i počet odkazů a úrovní v něm.

Čtvrtá verze (obr. 71) se již přibližuje finálnímu stavu, navigace je inspirována organizační strukturou klasické kartotéky – podle popisky záložky a její barvy si vyberu požadované téma, které je pak dále děleno na podkapitoly. Ústředním motivem se opět stává fotografie v hlavičce.

Touto verzí se potvrdil navigační systém celé prezentace a zbývalo jen doladit vzhled – fotografie v záhlaví byla dle mého názoru natolik výrazná, že rušila při čtení a studiu.

Proto vznikla **pátá verze** návrhu, ve které byla fotografie odstraněna a místo ní je použit černý pruh s lineárním gradientem – obr. 72.

Já sám jsem tuto verzi pokládal za finální, ale po další konzultaci s grafikem („... je to celé takové hrubé ostré, něco s tím ještě udělej!“) přišla na svět **šestá verze** – ta se od předchozí liší pouze zaoblenými rohy (obr.).



Obr. 71: Grafický návrh č.4



Obr. 72: Grafický návrh č.5

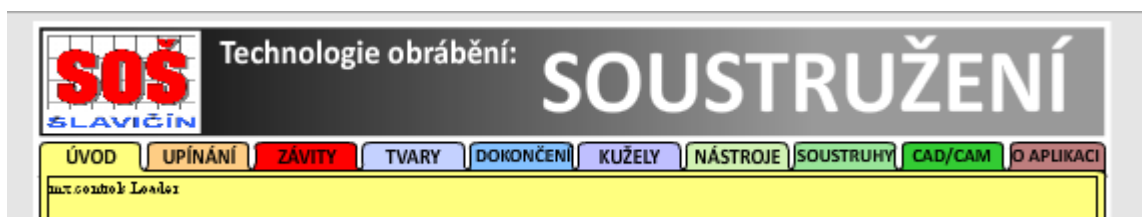
Nicméně i tato malá úprava vedla k několikanásobnému vylepšení vzhledu a mé naprosté spokojenosti (grafik si taky přestal stěžovat). Prohlásil jsem ji tedy za verzi finální a začal se věnovat dalším fázím.

2 Hlavní program

Základem celého projektu je scéna souboru prezentace. Scéna má rozměry 800x600 px: menší plocha by mohla působit stísněně a nečitelně, větší by se nemusela vměstnat na plochu monitoru - vycházím z toho, že většina studentů má stroje, jejichž nativní rozlišení začíná na 1024x768 pixelech.

Počet snímků za vteřinu (tzv FPS nebo také framerate) je nastaven na 15 – za rozumné minimum je považováno 12 snímků za sekundu, efektivní maximum pak 20 snímků. Obecně platí, že čím více snímků, tím jsou přechody pohybů jemnější a nepůsobí tak rušivě. Pokud však překročíme 20 snímků, uživatel již nerozezná rozdíl, zatímco se rapidně zvyšuje velikost souboru.

Do této scény jsou pak vloženy dvě samostatné skupiny: hlavička a skupina záložek – obr. 73.



Obr. 73: Scéna prezentace, nahoře hlavička, pod ní skupina záložek

2.1 Hlavička

Hlavička je uložena v samostatných vrstvách – *pozadi_hlavicka*, *logo*, *nadpisy*, *zvuk*. Všechny vrstvy mají jediný klíčový snímek a jsou rozprostřeny na celou využitou osu (jinými slovy – hlavička se během přehrávání vůbec nemění.)

2.2 Záložky

Zatímco hlavička je ve své podstatě statická, u záložek už to není tak docela pravda – i když nejsou použity dynamické prvky, vzhled tak vypadá.

Při tvorbě jsem použil několik barevných záložek. všechny mají stejný obdélníkový tvar a velikost, jenom poloha „oušek“ je specifická. Díky tomu se záložky překrývají a vidět je vždy pouze první, z ostatních právě jen ouška (obr).

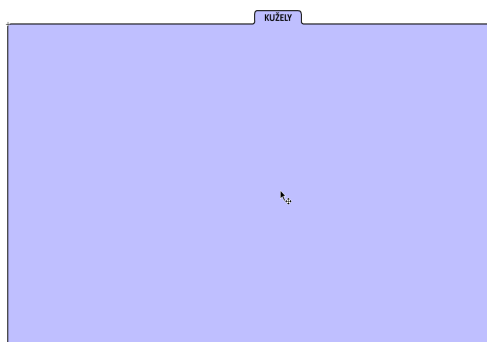
Přes záložky je pak položena ještě komponenta `mx.controls.Loader` – pomocí ní nahráváme do prezentace jiné, předem připravené soubory. V našem případě to jsou jednotlivé kapitoly encyklopedie.

Kapitoly jsou tvořeny tzv. Flash slide presentation soubory. Velmi jednoduše si je lze představit jako prezentaci (např. Powerpoint). Blíže jsou popsány v samostatné kapitole.

Celý princip změny záložky spočívá v tom, že po kliknutí na jméno záložky, zachytí toto kliknutí neviditelné tlačítko (viz níže), které přesměruje přehrávač na požadovaný snímek. V tomto snímku je v popředí požadovaná záložka, takže se mění barevný okraj. Navíc se v tomto snímku také mění vlastnosti Loaderu – nahraje .swf soubor příslušné kapitoly.

Záložky – barevné pozadí

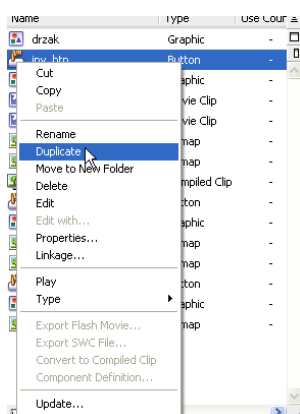
Při tvorbě pozadí záložek jsem potřeboval vytvořit tvar obdélníku, který je na horní hraně propojen s dalším. Výsledný objekt pak musel mít zaoblené všechny rohy (obr. 74.).



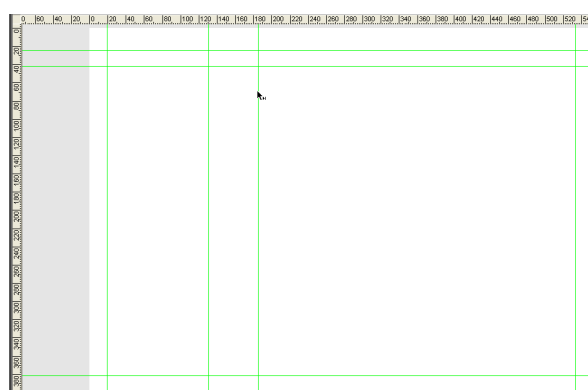
Obr. 74: Záložka

Jelikož flash umí zaoblit pouze vnější rohy, nebylo možno vytvořit tento tvar najednou, ale skládáním základních tvarů:

Krajní záložky jsem nakreslil tak, že jsem nejprve nastavil vodící čáry na požadované rozměry (obr. 76.), do nich nakreslil základní obdélník se zaoblenými rohy (obr. .). Tím došlo ke slítí stejnobarevných ploch do sebe. (Pozn.: Pokud bychom kreslili jinými barvami, plochy by se od sebe odečítaly). Do nové vrstvy pojmenované text, vložíme statické textové pole s názvem kapitoly – v tomto případě Úvod. Pro dokončení požadovaného vzhledu stačilo využít nástroje Inkoust, který vytvořil obrys.



Obr. 75: Duplikování symbolu



Obr. 76: Vodící čáry

Tím byla vytvořena levá krajní záložka, kterou jsem konvertoval do grafického symbolu. Pro vytvoření pravé krajní záložky šlo použít téhož postupu jako u levé, nebo si zjednodušit život: Symbol levé záložky jsem v knihovně prvků duplikoval pod jiným názvem (obr. .) Tento symbol jsem po otevření v editačním režimu zrcadlově otočil a změnil barvu výplně. Ve vrstvě text změním popis (O aplikaci).

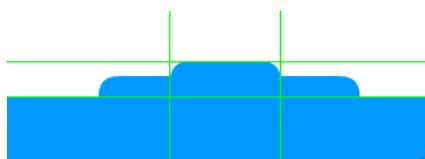
Důležité je po otočení zkontrolovat počáteční bod celého objektu – pro všechny záložky musí být ve stejném místě.

Při umístění záložky totiž stačí zadat stejné souřadnice počátečního bodu a máme jistotu perfektního překrytí. Pokud bychom chtěli záložky umístit pomocí nástroje Zarovnání (Align), narazíme na problémy způsobené právě nestejným umístěním oušek

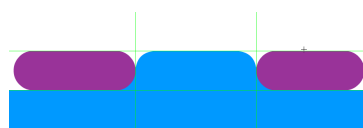
Po ukončení editace máme vytvořenu druhou záložku a přistoupíme ke tvorbě středových záložek.

Opět vytvoříme síť vodících čar podle požadovaných rozměrů, do ní zakreslíme hlavní obdélník, nad něj druhý který představuje ouško záložky (oba obdélníky mají nastaveno

zaoblení rohů). K nim přidáme ještě jeden obdélník, který bude tvořit zaoblení vnitřních rohů (zatím kreslíme vše v jediné barvě výplně bez obrysů – obr. 77.).



Obr. 77: Příprava na zakulacení vnitřních rohů



Obr. 79: Zakulacení různobarevnými obdélníky



Obr. 78: Hotové vnitřní zakulacení

Nyní vybereme jinou barvu výplně a nakreslíme obdélníky po obou stranách ouška (obr. 79). Tím dojde k odečtení nežádoucích ploch a vznikne zaoblení vnitřních rohů. Oba různobarevné obdélníky pak smažeme, inkoustem vytvoříme obrys a převedeme celý objekt na symbol. Vytvoříme novou vrstvu s názvem kapitoly (Upínání). Tím je první středová záložka hotova.

Další záložky vytváříme z první – duplikujeme záložku a přejdeme do režimu její editace. Dvojklikem na obrys jej celý označíme a smažeme. Pak posuneme vodící čáry na novou pozici, nástrojem výběr označíme ouško a pak jej přetáhneme na novou pozici (díky vodícím čarám se ouško lehce přichytí k požadované pozici). Vytvoříme obrys, změníme text popisky a záložka je hotova.

Postup opakujeme, pro vytvoření všech záložek (tedy ještě 7x).

Všechny záložky pak naskládáme na plochu, pomocí počátečního bodu je umístíme tak, aby se překrývaly a pomocí kontextové nabídky záložek (RMB) je seřadíme.

2.3 Tlačítka

Abychom mohli využívat interaktivitu, musíme použít tlačítka. Tato tlačítka umístíme přes názvy záložek a přiřadíme jim interakci pomocí chování.

Aby byly vidět názvy záložek a tlačítka nijak nerušila vzhled, vytvoříme je neviditelná – to znamená, že v tlačítku bude jediný snímek – Hit (obr. .), ve kterém nakreslíme libovolně velký obdélník s barevnou výplní, bez ohraničení.

Tím je zajištěno, že při přehrávání nebude tlačítko vidět, resp. bude zobrazen prázdný snímek. Prázdný snímek bude u tlačítka zobrazen i při najetí kurzoru a kliknutí. Jediný ukazatel existence tlačítka bude změna kurzoru – obr. 80.



Obr. 80: Tvar kurzoru nad tlačítkem

Po vytvoření symbolu tlačítka jej přetáhneme na plochu, umístíme nad první záložku, upravíme velikost aby překrýval celé ouško. Přes klávesu Alt pak takto upravené tlačítko překopírujeme nad ostatní záložky (nemusíme tak upravovat každé tlačítko zvlášť – jelikož jsou ouška záložek stejně velká, budou stejná i tlačítka).

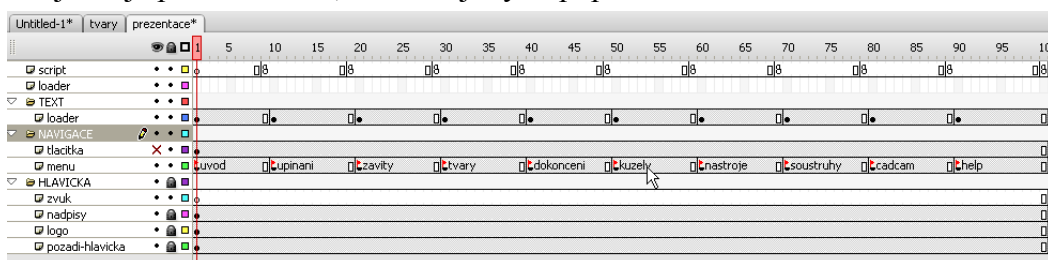
2.4 Časové osa

Pokud je scéna připravena, umístěny všechny komponenty, dotvoříme časovou osu, aby mohla zobrazovat změny při přepínání záložek.

Musíme tedy pomocí časové osy vytvořit všechny stavy záložek – vytvoříme deset stavů záložek. Jeden stav odpovídá jedné záložce v popředí. Všechny použité vrstvy protáhneme až na snímek č. 100.

Na každý stav tak připadne 10 snímků. I když by stačil snímek jediný, je lepší nechat si rezervu např. pro případné přidávání kódu AS apod.

Po protažení vytvoříme ve vrstvě záložek klíčové snímky vždy po 10 snímcích a pojmenujeme je podle záložek, které mají být v popředí – obr. 81.



Obr. 81: Pojmenované klíčové snímky

Projdeme jednotlivé klíčové snímky a příslušné záložky naaranžujeme do popředí (kontextová nabídka dané záložky vyvolaná RMB, příkaz Arrange/Bring to front – přenes dopředu).

Pokud nyní přejedeme jezdce časové osy přes všechny snímky, uvidíme, že se postupně mění záložky v popředí.

Chceme-li ověřit funkci jinak, klávesami Ctrl+Enter spustíme testovací přehrávání. Záložky budou měnit své pořadí cyklicky bez zastavení, což je pro nás nežádoucí – musíme zajistit zastavení přehrávání na žádaném úseku.

Přidáme tedy další vrstvu pro AS (lze použít některou ze stávajících, ale jak už jsem psal, samostatná vrstva přináší větší přehled). V této vrstvě vytvoříme další klíčové snímky (vždy snímek před klíčovým snímkem vrstvy záložek).

Do těchto klíčových snímků vložíme krátký kód:

```
stop ();
```

Tím zajistíme, že po přechodu přehrávače na 1. snímek stavu, dojde k přehrávání dalších 9 snímků a na posledním se přehrávač zastaví, dokud mu nedáme další podnět.

Po novém spuštění testovacího přehrávání se objeví scéna se záložkou Úvod v popředí, ale nic dalšího se neděje – to je způsobeno přehráním 9. snímku s kódem stop. Protože ještě nejsou nastavena tlačítka záložek, nelze se mezi nimi přepínat (kurzor ale tlačítko indikuje).

Zrušíme testovací přehrávání a nastavíme tlačítka – každému tlačítku musíme přiřadit takovou funkci, aby po kliknutí přesměroval přehrávač na dotyčný snímek. Tlačítkům tedy přiřadíme jednoduchý kód:

```
1 on (release) {
```

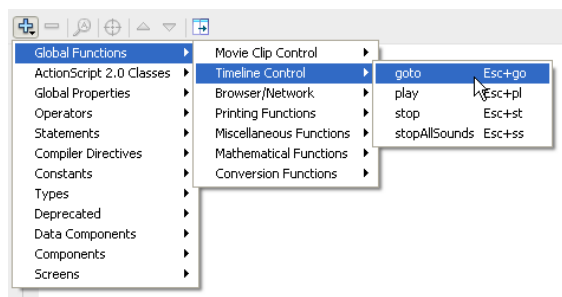
po uvolnění tlačítka

```
2 gotoAndPlay("uvod");
```

přejdi na snímek s názvem „úvod“ a přehraj jej. Každé tlačítko bude samozřejmě mít jiný cílový snímek.

```
3 }
```

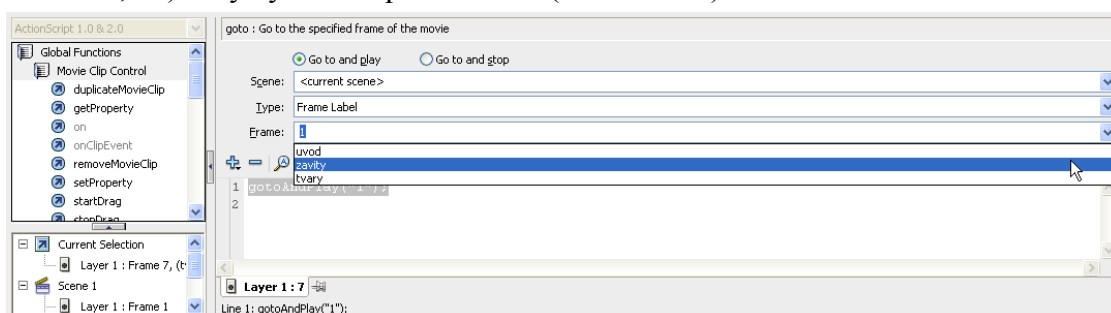
Kód vložíme pomocí asistenta: přes tlačítko + vložíme kód z nabídky Global function / Timeline control / GoTo (obr. 82.).



Obr. 82: As - Asistent

Při zapnutí asistentovi, se objeví nová nabídka (obr. 83.), ve které nastavujeme:

- přehrávač má po přechodu na nový snímek zastavit nebo jej rovnou začít přehrávat (Go to and stop/Go to and play)
- ve které scéně se požadovaný snímek nachází (Scene)
- podle jakého atributu chceme náš snímek hledat (podle čísla, jména, následující snímek, ...) - my vybereme podle názvu (Frame label)



Obr. 83: AS - Nastavení atributů pomocí asistenta

- a nakonec atribut snímku, podle toho, co jsme zvolili v předchozím kroku. My vybereme snímek s názvem úvod (obr. 83.)

Tento postup zopakujeme pro každé tlačítko záložky, samozřejmě s příslušným odkazem. Pokud nyní spustíme testovací přehrávání, prezentace se zastaví na snímku úvod, my však můžeme libovolně přepínat mezi záložkami přesto, že nemají ještě žádný obsah.

2.5 Loader

Aby byla scéna kompletně připravená, zbývá jen dodat před vrstvu se záložkami výše zmiňovaný loader (mx.controls.Loader). Ten nalezneme v paletce komponent, složka

uživatelské rozhraní (user interface). Loader umístíme na požadované místo, nastavíme jeho velikost.

Jak jsem již dříve napsal, loader užíváme k přehrání jiných .swf souborů v aktuálně přehrávaném. K tomu účelu lze loader dále uzpůsobit – v záložce inspektora komponenty (Component inspector) nastavujeme další vlastnosti (obr. 84)

Autoload – automatické načtení obsahu. Pokud zvolíme false (nenahrávat automaticky), museli bychom použít ještě další prvek, kterým bychom dali přehrávači příkaz k nahrání souboru.

ContentPath – cesta k obsahu. Zadáváme relativní cestu k .swf souboru, který chceme přehrát.

ScaleContent – automatická změna měřítka obsahu. Je-li zapnuta (true), přizpůsobuje se obsah velikosti okna, v opačném případě se okno přizpůsobí velikosti přehrávaného obsahu.

Enabled – povoleno. Je-li prvek povolen (true) funguje jak má. Pokud jej zakážeme, prvek se vůbec nebude zobrazovat – přehrávač jej bude ignorovat.

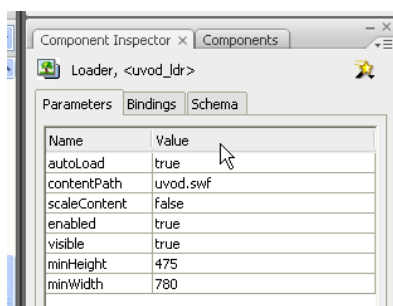
Min Height, Min Width – minimální výška, minimální šířka. Tyto položky mají význam pouze při ScaleContent:false.

Abychom nemuseli složitě programovat, použijeme loader podle následujícího schématu: po přesunu přehrávače na novou pozici se musí změnit i obsah loaderu. Stačí tedy rozdělit vrstvu loaderu stejně jako je rozdělena vrstva záložek. Při přehrávání se tedy změní obsah loaderu a pořadí záložek v jeho pozadí.

Jelikož loader leží v samostatné vrstvě, bude práce o to jednodušší – ve stejném místě jako u záložek vložíme klíčové snímky, ve kterých pak budeme měnit obsah loaderu.

Pokud nyní vyzkoušíme testovací přehrávání, zjistíme, že nedošlo k žádným změnám – záložky se sice mění jak potřebujeme, ale není vidět žádný obsah záložek. Ony jej samozřejmě mají – vždyť jsme přes ně položili loader. Ten ale ještě nemá určený obsah, takže se zobrazuje, ovšem jako prázdný snímek.

Nyní musíme projít jednotlivé klíčové snímky a nastavit v nich cestu k předem připraveným .swf souborům (jednotlivým kapitolám). To provedeme tak, že vybereme klíčový snímek (např. úvod), v něm označíme prvek mx.controls.Loader a v inspektoru komponenty upravíme položku Content Path: uvod.swf (obr. 84).

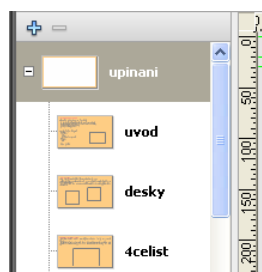


Obr. 84: Vlastnosti Loaderu

Tento postup opět zopakujeme pro všechny další klíčové snímky. Pokud otestujeme přehrání nyní, hlavní scéna již funguje, jak má – po přepnutí záložky se změní i obsah loaderu.

3 Prezentace obrazovek

Celý systém prezentace je založen na vztazích mezi obrazovkami – nejprve jsou rodičovské obrazovky (my budeme používat jedinou rodičovskou obrazovku – tu hlavní), jim podřízené jsou obrazovky dceřinné (obr. 85). Ty „dědí“ vlastnosti svých rodičů a navzájem jsou si rovnocenné (říkáme, že jsou na stejné hladině).



Obr. 85: Rodičovské
a dceřinné obrazovky

3.1 Rodičovská obrazovka

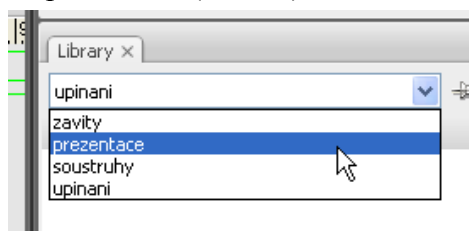
Výchozí název je „presentation“, poklepnáním na název jej ale lze měnit. Na rodičovské obrazovce je nejdůležitější, že cokoliiv na ní vytvoříme nebo provedeme, změní se i ve všech dceřinných obrazovkách.

Toho využijeme při tvorbě navigačního systému uvnitř obrazovek – ta se skládá z navigačního proužku v horní části obrazovky a ze šipek ve spodní části (funkce předchozí a následující stránka).

Rodičovská obrazovka má pozadí ve stejné barvě jako příslušná záložka a na ní je teprve umístěn bílý obdélník jako podklad pro text. Samozřejmě by šlo použít přímo bílý podklad rodičovské obrazovky, ale pak by nebylo možné vytvořit zaoblení rohů. Výsledek by pak mohl působit rušivě. Obdélník není roztažen po celé ploše, ale nahoře odsazen – zde bude navigační proužek.

Ten vytvoříme jednoduše tak, že pomocí statického textu vytvoříme popisky podkapitol (např. pro kapitolu Upínání jsou to: Úvod, Upínací desky, Upínací trny, Opěrky, Kombinované způsoby). Ty umístíme do přibližné polohy a vyrovnáme nástrojem align (obr. 11.). K nim pak ještě přidáme dělítko – krátkou čáru vytvořenou nástrojem přímka. Opět umístíme do přibližné polohy a vyrovnáme nástrojem align.

Nyní máme hotový vzhled, ale stále chybí tlačítka. Opět využijeme neviditelných tlačítek – je jedno, zda je vytvoříme znovu (jejich tvorba je jednoduchá), nebo je „přetáhneme“ ze souboru prezentace. Máme-li otevřeny oba soubory, stačí si v seznamu knihoven vybrat místo souboru upínání soubor prezentace (obr. 86).



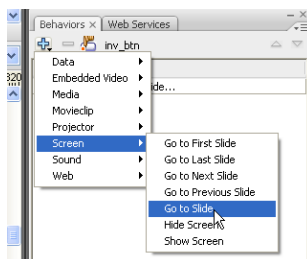
Obr. 86: Seznam knihoven

Po přetažení na plochu se symbol neviditelného tlačítka automaticky uloží do knihovny. Připravíme si na ploše požadovaný počet tlačítek a pomocí nástroje Volná transformace je umístíme nad popisky a upravíme jejich velikost – obr. 87.



Obr. 87: Volná transformace

Tlačítkům nyní přiřadíme funkci přechodu na jinou obrazovku. Nejrychlejší je použít Chování (Behaviors) – paletka Behaviors, nabídka: Add behavior/Screen/Go to screen (obr. 88). V následujícím dialogu stačí už jen vybrat vhodnou obrazovku, na kterou má tlačítko přepnout.



Obr. 88: Přechod na konkrétní obrazovku

Tím dojde k vložení AS kódu do tlačítka:

```
1 on (release) {
2 // GoTo Screen behavior
```

řetězec // označuje poznámku, která nemá na kód vliv – v tomto případě poznámka oznamuje, že začíná kód generovaný nástrojem chování a že tento kód plní funkci Jdi na (GoTo).

```
3     if((this.kombinace_1 != undefined) && (this.kombinace_1 !=
4         null))
5         {
6             var screen = null;
7             var target = this;
```

Var je kódové slovo pro přiřazení hodnoty nové proměnné. V následujícím kódu jsou uvedeny možnosti které mohou nastat (přehrávač je na pozici, kterou odkazuje tlačítko; přehrávač je na jiné pozici) a co se při nich má dít (nic; přesun na požadovanou pozici):

```
7         while((screen == null) && (target != undefined) &&
8             (target != null))
9         {
10            if(target instanceof mx.screens.Screen)
11            {
12                screen = target;
13            }
14            else
15            {
16                target = target._parent;
17            }
18        }
19        if(screen instanceof mx.screens.Slide)
20        {
21            screen.gotoSlide(this.kombinace_1);
22        }
```

```

23// End GoTo Screen behavior
24 }

```

Tento postup musíme zopakovat pro všechna tlačítka. Pro navigaci v rámci obrazovek budeme používat i šipky pro posun po jednotlivých stránkách. Ty ale nemůžeme použít na rodičovské obrazovce, protože nebudou vždy stejné na všech stránkách – na první stránce je šipka pouze vpravo (nejde přejít na předchozí stránku), stejně tak na poslední obrazovce nemůže být šipka vpravo – nejde přejít na další stránku.

3.2 Dceřiné obrazovky

Všechny použité dceřiné obrazovky jsou v hierarchii stejně důležité – jsou na stejné úrovni – to je patrné již z grafického znázornění skladby celé prezentace (obr. .). Toto uspořádání nám ulehčuje práci při tvorbě navigace po jednotlivých stránkách – stačí použít jednoduché tlačítko s akcí „přesun na další stránku“. Pořadí stránek odpovídá pořadí ve stromu obrazovek.

Vytvoření tlačítka je triviální záležitost – do snímku Up vložíme textové pole se dvěma znaky > a ve snímku Hit je překryjeme obdélníkem libovolné barvy (tím určujeme aktivní plochu). Druhé tlačítko vznikne duplikací prvního a úpravou, která spočívá v otevření textového pole a nahrazení znaků > dvěma znaky <.

Symbol by šlo po duplikaci i rotovat o 180°. Je pravda, že bychom tím šetřili velikost souboru, ale tato úspora je zanedbatelná, nehledě na fakt, že při rotaci by se posunul i klíčový bod symbolu (střed řádku písma totiž není v geometrickém středu písmen). Z tohoto důvodu jsem tedy zvolil první způsob.

K oběma tlačítkům musí být i příslušný program: pro tlačítko „Následující obrazovka“ bude vypadat takto:

```

1 on (release) {

```

Po uvolnění tlačítka pokračuje vykonávání skriptu

```

2 // GoTo Next Screen behavior

```

Poznámka o zdroji kódu – generován nástrojem chování s funkcí „přejdi na další obrazovku“.

```

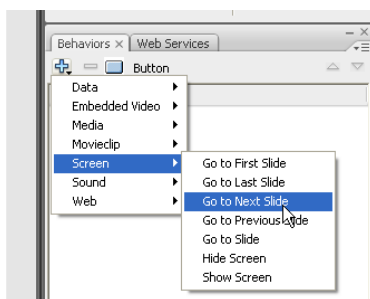
3     var screen = null;
4     var target = this;

```

Definice proměnných, následuje samotný program – pomocí příkazů if rozlišíme možné stavy, a provedeme akci:


```
5     while((screen == null) && (target != undefined) && (target !=
      null))
6     {
7         if(target instanceof mx.screens.Screen)
8         {
9             screen = target;
10        }
11        else
12        {
13            target = target._parent;
14        }
15    }
16    if(screen instanceof mx.screens.Slide)
17    {
18        screen.rootSlide.currentSlide.gotoNextSlide();
19    }
20// End GoTo Next Screen behavior
21}
```

Kód pro obě tlačítka vkládáme opět přes nástroj chování (Add Behavior / Screen / Go to next slide, resp. Go to previous slide) – obr. 89.



Obr. 89: Chování - Další
obrazovka

Navigační šipky byly vloženy nejprve do snímku č. 2, odkud byly nástrojem „Kopírovat do stejného místa“ přeneseny na všechny ostatní. Z prvního a posledního snímku jsem samozřejmě nežádoucí šipku vymazal.

Kromě navigace se v obrazovkách nachází samotný informační obsah: text, obrázky, animace a videosekvence - všechny prvky jsou popsány v samostatných kapitolách.

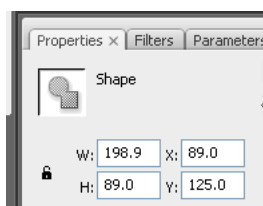
4 Textová část

Textová část, přestože je v prezentaci nejvíce zastoupena, byla programově velmi jednoduchá, i když časově asi nejnáročnější.

4.1 Text – informační obsah

Při práci jsem využíval statické textové pole, které jsem pomocí pravítek a vodících čar na rodičovské obrazovce umístil do první dceřiné obrazovky, ze které jsem je pak kopíroval na další obrazovky pomocí příkazu Kopírovat – Vložit na stejné místo (Copy – Paste in Place). Polohu vodítek jsem si samozřejmě poznačil pro použití v další sadě obrazovek – jiných kapitolách. Tím byla zajištěna stejná poloha všech textových polí a odstraněno nepříjemné „poskakování textu při přepínání jednotlivých stránek.

Další možností, jak zajistit stále stejnou polohu objektu je použití souřadného systému – zadáním číselné hodnoty pro polohu levého horního rohu a velikosti objektu v ose X a Y je zaručeno minimálně stejně přesné umístění – obr. 90.

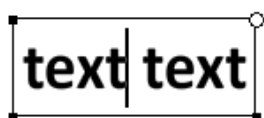


Obr. 90: Velikost a umístění prvku

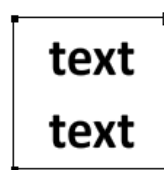
Znovu podotýkám, že oba tyto způsoby stačí použít v dané kapitole jen jednou na první dceřinou obrazovku a pak využívat kopírování na stejné místo.

Velikost textového pole je ošetřena pouze horizontálně – výška se automaticky přizpůsobuje obsahu – je-li pole prázdné, výška se snižuje (levý horní roh zůstává na své pozici). Naopak při postupném plnění znaky se prodlužuje.

Šíře se může měnit dynamicky – stačí že kliknutím vytvoříme textové pole, nebo je nastavena pevně: po vytvoření textového pole se v jeho pravé části objeví prázdné kolečko (pouze v režimu editace textu) – obr. . Pokud jej uchopíme a přetáhneme jinam, změní se ve čtvereček, který signalizuje pevně nastavenou šíři – obr. .. Chceme-li tuto funkci znova vypnout, stačí na čtvereček dvakrát poklepat.



Obr. 91: Texts dynamickou
šířkou



Obr. 92: Text s pevnou
šířkou

4.2 Text – navigace

Pro popisky navigačních tlačítek bylo opět využito statických textových polí. Přes ně jsou v další vrstvě neviditelná tlačítka (viz kapitola Navigace). Textová pole jsou umístěna v samostatné vrstvě pro snazší úpravu a organizaci.

5 Obrázky

Ve své prezentaci používám dva druhy obrázků – ty, které vytváříme přímo ve Flashi (pomocí nástrojů pro kreslení) nebo ty, které do prezentace importujeme.

5.1 Obrázky z Flashe

Pro ukázkou si popíšeme tvorbu několika obrázků

Závitový nůž

V prezentaci je použit v kapitole Závity – Závitové nože. Tento obrázek je příkladem jednoduchého postupu – nevyžaduje víc vrstev, nepoužíváme žádné další symboly (obr. 95.):

- Nejprve vytvoříme tři stejnobarevné obdélníky bez obrysu (obr. .). Z jednoho vytvoříme špičku nástroje spojením dvou vrcholů, z dalšího vytvoříme zakončení nože (sražení) – obr. 93.



Obr. 93: Příprava nože

- Poté všechny tři obdélníky spojíme do jediného kusu a vytvoříme obrys.
- Nakreslíme přímkou pro označení sražení na těle nože a přímkou s kružnicí pro oddělení břitové destičky – obr. 94.



Obr. 94: Obrisy nože

- d) Změníme barvu destičky – obr. 95.



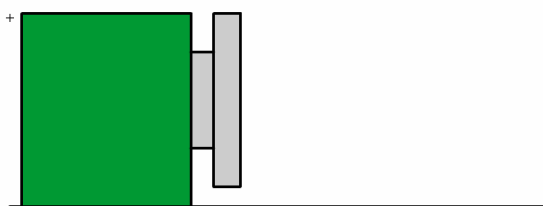
Obr. 95: Dokončený závitový nůž

- e) celý nůž označíme a konvertujeme do symbolu

Čelní soustruh

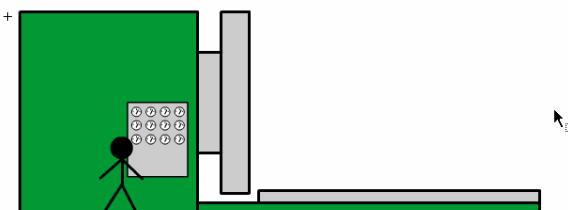
Jelikož se jedná o složitější obrázek, je rozdělen do několika vrstev pro snazší orientaci: zem, vřeteník, lože, ovládání, dělník, obrobek, suport a text. Během práce automaticky převádíme právě vytvořené prvky na symboly! Postup:

- Do vrstvy zem vložíme rovnou přímkou – ta představuje podlahu
- Ve vrstvě *vřeteník* nakreslíme obdélník představující vřeteník a k němu další dva obdélníky – vřeteno a upínací deska – obr. 96.



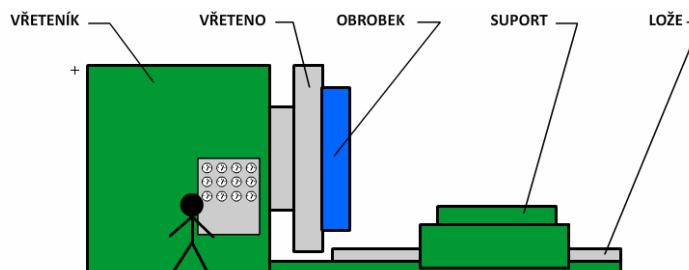
Obr. 96: Vřeteno a upínací deska

- Do vrstvy *ovádání* vložíme zjednodušený ovládací panel a do vrstvy *dělník* postavíme dělníka (složena z pěti čar a kružnice bez výplně) pro orientační srovnání velikosti člověka a soustruhu – obr. 97.



Obr. 97: Ovládací panel a dělník

- d) Obdélník ve vrstvě *obrobek* bude představovat obrobek
- e) Nakreslíme obdélník představující kus do příslušné vrstvy, stejně tak do vrstvy *suport* nakreslíme další dva spojené obdélníky, představující suport.
- f) Celý obrázek dokončíme přidáním popisek do vrstvy *text* – obr. 98.



Obr. 98: Kompletní obrázek čelního soustruhu

Na úplný závěr převedeme celý výtvar (stroj s popisky) na grafický symbol.

Kresby vytvořené ve Flashi rozhodně nejsou uměleckými díly (a o těch mých to platí dvojnásob), mají jediný úkol: zjednodušeně něco představit, ukázat. Pokud by kdokoliv chtěl ve Flashi tvořit nádherné ilustrace, pohoří – k tomu jsou určeny jiné programy.

Obecně platí, že v závislosti na situaci, je vhodné maximálně zjednodušovat – on sice soustruh (budu-li se držet svého tématu) vykreslený do posledního šroubečku je určitě nádherný, ale rozhodně zbytečný – proč trávit celé dny (obavám se dokonce, že v případě onoho soustruhu by se jednalo spíše o týdny) prací, když uživateli stačí pro ilustraci hrubý obrys s několika popisky?

5.2 Obrázky importované

Kromě dvou obrázků z cizích zdrojů (jedná se o fotografii stolního soustruhu a sken soukolí pro nastavení posuvu při výrobě závitu) jsou všechny obrázky, animace a videa mým vlastním výtvozem.

Obrázky jsem buď vyfotil v reálu – buď fotoaparátem OLYMPUS FE-150 (5Mpixelů) nebo kamerou SONY DCR-SR32E (3Mpixely) nebo vytvořil v jiných programech (Solidworks 2007) a nasnímal pomocí XnView (nástroj snímání obrazovky).

Takto získané snímky jsem pak importoval do jednotlivých prezentací, nebo nahrával do prezentace pomocí komponenty loader. Tento způsob má velkou výhodu v tom, že pokud budu chtít v budoucnu změnit nějaký obrázek, nemusím zasahovat do celé prezentace, stačí prostě přepsat původní soubor obrázku.

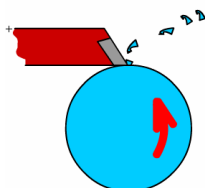
Jelikož je import obrázku popsán v samostatné kapitole, nebudu jej tady popisovat, snad jen dodám, že komprese importovaných obrázků je minimální – na prvním místě byla kvalita.

6 Animace

O animacích lze říct totéž co o obrázcích – mohou být opravdu kvalitní, ale autor na nich stráví polovinu života. Proto jsem své animace maximálně zjednodušoval (pevně věřím, že ne na úkor srozumitelnosti).

6.1 Tangenciální nůž

Pro větší představu o tom, jak fungují animace, opět uvedu popis postupu při jednoduché animaci tangenciálního nože. Tato animace se skládá ze tří objektů: rotujícího obrobku (kružnice s modrou výplní), nože a třísek. Obrobek se otáčí na místě, nůž se k němu pomalu přibližuje a při dotyku začnou odletovat třísky – obr.. Postup:



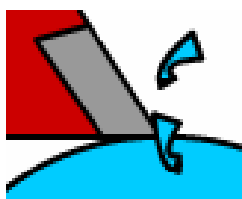
Obr. 99: Animace
tangenciálního nože

- vytvoříme MC symbol „rotující_kus“, ve kterém do jedné vrstvy vložíme kružnici s modrou výplní a obrysem černou linkou. V druhé vrstvě umístíme symbol šipky. Obě vrstvy protáhneme na snímek č.30. Vrstva s kusem je hotová, vrstvu se šipkou musíme animovat – na poslední snímek vložíme další klíčový snímek a pro první snímek určíme Tween: motion s atributem Rotate: CCW (CounterClockWise – proti směru hodinových ručiček) , 2 times (počet rotací během animace).
- vytvoříme nový symbol, do první vrstvy (*obrobek*) vložíme právě vytvořený MC rotující kus a prodloužíme jej na délku 75 snímků
- do nové vrstvy (*nuz*)přidáme předem připravený grafický symbol nože (obr. .) a prodloužíme ji na 75 snímků.

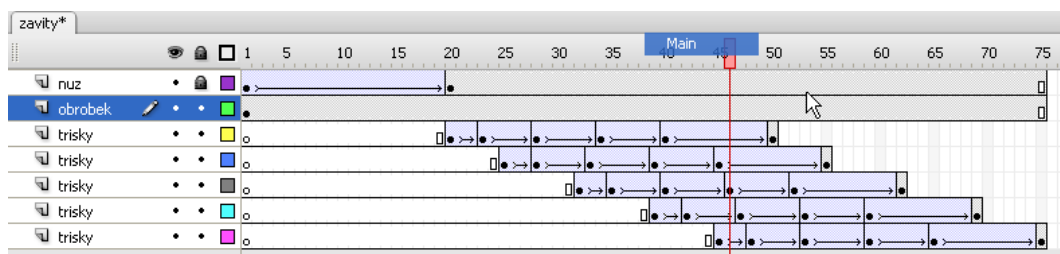


Obr. 100: Tangenciální nůž

- d) do 20. snímku vrstvy *nuz* vložíme klíčový snímek, ve kterém se nůž dotýká obrobku.
- e) mezi 1 a 20 snímek vložíme animaci (Tween: motion), ve které se nůž ze základní polohy přiblíží k obrobku a zůstane u něj.
- f) pod vrstvu *obrobek* vložíme 5 dalších vrstev (*triska_1* - 5), do kterých následně umístíme animaci odletujících třísek. Vrstvy třísek jsou pod obrobkem umístěny záměrně – Flash totiž zobrazuje objekty podle pořadí vrstev, ve kterých jsou umístěny. Kdybych umístil třísky nad obrobek, vznikl by vypadálo přinejmenším divně (obr. 101)

Obr. 101: Tříška
před obrobkem

- g) do vrstvy *triska_1* vložíme připravený grafický symbol třísky a vytvoříme jeho animaci, ve které odletuje od nože a otáčí se při ní.
- h) snímky z vrstvy *triska_1* překopírujeme do ostatních vrstev *triska_x*. Tyto snímky musíme vzájemně posunout (obr. 102), takže vzniká dojem, že od nástroje odlétá celý roj třísek.



Obr. 102: Časová osa animace

7 Videosekvence

Všechny videosekvence byly snímány výše zmíněnou kamerou SONY DCR-SR32E přímo na dílenských učebnách naší školy (SOŠ Slavičín) nebo v prostorách firmy ZKL Praha, pobočka Rokytnice.

Všechna videa byla sestříhána pomocí programu VirtualDUB MPEG-2 1.6.15 ([20]) . Pro komprimaci jsem stáhl demoverzi programu Sorenson Squeezer ([18]), kterou doporučuje i firma Macromedia a Adobe (dřívější a nynější vydavatel programu Flash). Program obsahuje nejen kodek, ale také pracovní prostředí pro komprimaci. Bohužel se ukázalo, že systém práce v něm je (alespoň pro mne) naprosto nepochopitelný.

Naštěstí se při instalaci kodeku nainstaloval odkaz i do mého oblíbeného VirtualDUBu, takže jsem se do něj s radostí vrátil a provedl stříh i komprimaci najednou. Musím říct, že použitý kodek mě více než mile překvapil – např. jedno z původních videí o velikosti 40MB se stříhem a komprimací zmenšilo na neuvěřitelných 1,6MB!

Abych předešel narážkám na velikost odstřížených částí podotýkám, že z původního souboru bylo použito cca 95% materiálu.

Videa pak byla do prezentací vkládána přes komponentu loader. Ta je popsána v samostatné kapitole, takže ji nebudu popisovat znovu.

8 Audio

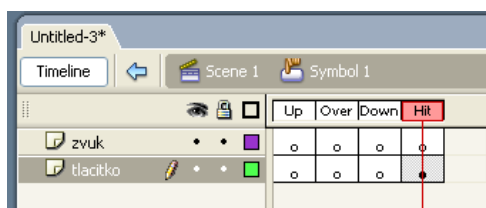
Zvuků je v prezentaci pohříchu málo – přesto že jsem plánoval do hlavního souboru importovat alespoň jednu skladbu v .mp3 a ovládat ji tlačítky „přehrát“ a „zastavit“, nakonec jsem od něj ustoupil – vyzkoušel jsem systém učení našich studentů a bylo by to zbytečné (pokud můžou, vypnou jakýkoliv zvuk a pustí si svou „hudbu“ - přes reproduktory nebo alespoň přes sluchátka).

U videí jsem zvuk záměrně vypustil, vždyť kdo by chtěl poslouchat monotónní hučení stroje?

Jediné zvuky, které zůstaly, jsou zvuky tlačítek. Postup zapojení do prezentace:

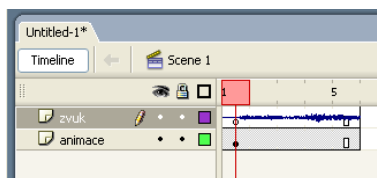
- Nejprve je potřeba zvuky získat – buď si je člověk vyrobí sám (časově náročné), nebo použije zvuk vyrobený někým jiným. Na internetu existuje velké množství serverů, které nabízejí volně šiřitelné zvuk. Z nejkvalitnějších jmenujme alespoň www.findsounds.com nebo freesound.iua.upf.edu.

- Z těchto serverů jsem si tedy stáhl dva zvuky kliknutí – jeden se ozve po najetí na tlačítko a druhý při samotném kliknutí
- Oba zvuky pak importujeme do knihovny symbolů (File/Import/Import to library)
- Nyní otevřeme symbol neviditelného tlačítka a do časové vrstvy vložíme novou vrstvu *zvuk*. V ní vytvoříme každému stavu prázdný klíčový snímek (obr. 103.).



Obr. 103: Klíčové snímky neviditelného tlačítka

-
- Nejprve označíme snímek over a na plochu přetáhneme z knihovny zvuk over. Že je zvuk použit je indikováno přímo ve snímku časové osy – objeví se v něm naznačená zvuková oscilace – obr. 104.



Obr. 104: Indikace zvuku ve snímku

- Do snímku down přetáhneme zvuk click. Opět je indikován oscilační křivkou.
- První zvuk (over) se přehraje, pokud přes tlačítko přejedeme kurzorem a druhý zvuk (click) bude hrát při stisknutí tlačítka.

9 Webové služby, formuláře

Tento postup lze využít například při odesílání dotazníků, sbírání výsledků testů apod.

Webová služba funguje na základě spojení poskytovatele služby (subjekt, který zajišťuje provoz služby po HW i SW stránce), registru služby (místo, kde jsou uloženy informace o webových službách, jejich poskytovatelích, informace o připojeních ...) a uživatele (tedy

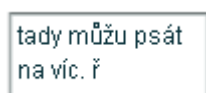
toho, kdo chce službu využívat.

Jedná se o technologii pro spojení prezentace s autorem. Nejprve vyplní uživatel formulář, který je pak přenesen pomocí webové služby např. pomocí zprávy elektronické služby ([21] nebo [22]).

Pokud chceme tyto služby využívat, musíme mít nejprve poskytovatele a registr služeb. Poskytovatelů WSDL je dnes již přehršel, mnoho z nich nabízí část svých služeb i zdarma (například [13]). Problém nastává s registrem služeb – vytvoření vyžaduje dobré znalosti XML a dalších technologií, které já bohužel neovládám (zatím!). Pomohla mi však samotná firma Adobe, která na svých server poskytuje registr služeb pro zkušební účely (výborně popsáno v [17], Lekce 10). Registr se nachází na adrese <http://www.forta.com/misc/flash2004tfts/email.cfc?WSDL> a i když má jen omezené možnosti, pro naše účely stačí.

Pro využití těchto služeb je potřeba nejprve sestavit požadovaný formulář – k tomu využíváme komponenty:

- Popis (Label) – obr. 106. – pro označení jednotlivých položek formuláře. V podstatě statický text
- Tlačítko (Button) – obr. 107. – tlačítku přiřadíme funkci pro odeslání celého formuláře
- Vstupní text (TextInput) – obr. 108. – pole pro vkládání textu – předmět zprávy, emailová adresa, ...
- Textová oblast (TextArea) – obr. 105. – podobně jako vstupní text, ale oblast je víceřádková – pro text zprávy apod.
- Číselník (NumericStepper) – obr. 110. – obsahuje klikací nabídku, ze které můžeme vybírat čísla – např. doba studia
- Roletková nabídka (Combobox) – obr. 109. – funguje podobně jako číselník, je však určen pro obecné řetězce znaků
- Zaškrťovací políčko (CheckBox) – obr. 103. – po kliknutí na políčko se označí, slouží např. pro výběr z několika možností, např. při otázce Jste: student/ vyučující?



Obr. 105: Textová oblast (TextArea)



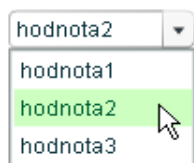
Obr. 106: Popis (Label)



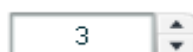
Obr. 107: Tlačítko (Button)



Obr. 108: Vstupní text (TextInput)



Obr. 109: Roletková nabídka (Combobox)

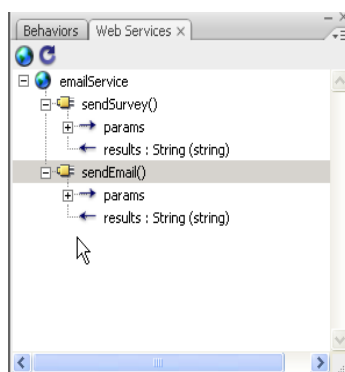


Obr. 110: Číselník (NumericStepper)



Obr. 111: Zaškrťovací políčko (CheckBox)

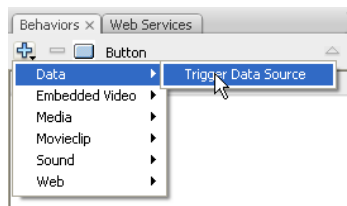
Po sestavení formuláře, je potřeba definovat WSDL – v paletce Web Services přidáme naši službu. Je-li přidání v pořádku, v seznamu služeb se objeví nová služba s rozevíracím stromem vlastností (obr. 112). Z nich je patrné, jaké služby dotyčná služba nabízí: jakou komunikaci (směr šipek), druh řetězců (string, boolean, ..) a názvy jednotlivých položek (EmailTo, EmailFrom,...).



Obr. 112: Paletka webových služeb, zobrazeny dostupné služby registru

Při kliknutí na položku můžeme v jejích vlastnostech přiřadit položce některý z objektů ve formuláři. Předpokladem je samozřejmě pojmenování instancí, protože právě názvy instancí je rozlišují.

Jako třetíčkou na dortu přiřadíme tlačítku funkci Odeslat pomocí chování (Add behavior / Data / Data trigger – obr.).



Obr. 113: Přidání chování -
odeslat data

10 Publikování

Při publikování jsem všechny části prezentace publikoval do Flash formátu – tzn. .swf souborů. Důvodem bylo další používání těchto podkapitol v hlavním souboru, při použití jiných formátů by nemusely fungovat ovládací prvky apod. Postup publikování je popsán v kapitole 2.8 Publishing.

Dalším krokem byla příprava výsledného CD – do složky jsem si připravil několik dalších adresářů:

- Diplomová práce – text
- Diplomová práce – program
- Flash Player
- Pluginy Flash playeru pro prohlížeče
- Demoverze Flash CS3

Pomocí Dreamweaveru (vytvořím jednoduchou webovou stránku, která bude obsahovat odkazy na jednotlivé soubory. Pomocí jednoduchého textového editoru vytvořím soubor autorun inf s následujícím kódem :

```
1 [autorun]
2 open=index.html
3 icon=icon.gif
```

Tento kód zaručí automatické spuštění prezentace, resp. úvodní stránky (rozcestníku) při vložení média.

ZÁVĚR

Cílem práce bylo srovnat možnosti e-learningu, porovnat jejich silné a slabé stránky a na jejich základě vybrat nejvhodnější způsob tvorby vlastní e-learningové pomůcky – Multimediální encyklopedie obrábění.

Teoretická část je rozdělena na dvě sekce – v první (E-learning a multimediální výchova) jsou popsány nejen jednotlivé, běžně používané způsoby e-learningu, ale zároveň vysvětleny jejich hlavní výhody a nevýhody. V závěru pak autor podle požadavků, které jsou kladeny na výsledný produkt postupně hodnotí kvalitu jednotlivých řešení a vybírá pro nás nejvhodnější způsob. Tím je zřejmě tvorba vlastní prezentace pomocí programu Flash.

Ve druhé sekci je již popisován samotný program – nejprve pracovní filozofie, posléze jsou stručně popsány i základní nástroje a pracovní prostředí.

V praktické části jsou pak podrobně rozvedeny další postupy – jak probíhala fáze návrhu, jaké způsoby byly pro realizaci zvoleny a jejich konkrétní postupy při tvorbě obrázků animací apod., takže i člověk, který doposud neměl s programem tu čest, si může udělat obrázek o stylu práce.

Přesto, že jsou cíle splněny, rozhodně nelze prezentaci označit za hotovou (podle kapitoly 1.2.3 Shrnutí se celý projekt pohybuje ve fázi 1.). Proto by autor rozhodně neměl svou práci odložit, ale naopak se jí dále věnovat tak, aby prošla i 2. a 3. fází a stala se tak plnohodnotnou učební pomůckou nejen pro několik konkrétních studentů, ale pro všechny zájemce o danou problematiku.

Jako největší přínos tohoto díla vidím dobrou interaktivitu mezi autorem (učitelem) a uživateli (studenty). Tento vztah již umožnil sbírat první poznatky a připomínky ze strany studentů, takže v nejbližší době bude stávající verze upravena (po obsahové stránce).

O užitečnosti programu svědčí také to, že jej má a používá přes 90% studentů 3. ročníku, kterým je určen, ale také asi 15% ostatních studentů, převážně z vyšších ročníků (3. a 4. ročníky odborných maturitních oborů a 3. ročníky učebních oborů).

Zjevnou nevýhodou je absence vyhledávání v obsahu, i když implementace zkušebních otázek by celému produktu dodalo jiný rozměr. Další verze by tedy měla umožňovat alespoň základní vyhledávání.

RESUME

Purposes of this work was level possibilities e-learningu, compare their strong and weak pages and on their basis choose optimal way of production personal e-learningove setout – multimedia encyclopedia cutting.

Theoretic part is fission on two branch – in first (E-learning and multimedia upbringing) they are described not only in singles, in common use manners e-learningu, but at the same time explication their chief benefits and disadvantage. In the end the author then according to requirements, that are laying on resulting product step by step evaluative making single solving and selection for us optimal way. Thereby come through production personal presence by the help of programme Flash.

In second branch is already describing himself programme – first working philosophy, finally be brief described and fundamental tools and working environment.

In the practical parts they are then in detail expansion next progress – how proceed design stage, what manners go in for realization election and their concrete progress at production pictures animation and so on., so that and man, which till now didn't have with programme here honour, yourself be able to do picture about style work.

Nevertheless, that being purposes executed trust, definitely it is impossible presence label as ready (according to chaps 1.2.3 summary with whole project move in phase 1.). That is why would author definitely didn't have of his work unrobe, but on the contrary with her further inscribe to so, to skirt and 2. and 3. phase and become so full-value instructional setout not only for several concrete student, but for all interested person about laid problems.

Like biggest contribution hereof works see good interaktivitu among author (teacher) and user (student). This relation already make collect first piece of knowledge and reminder edgewise student, so that no further particulas will current version modified (after contained pages).

About serviceability programme witness too it, that she's and using over 90% student of 3. vintage, that is identified, also perhaps 15% of others student, largely from higher vintage (3. and 4. vintage of revolting leaving examination branch and 3. vintage of instructional branch).

Apparent disadvantage is absence searching in content, though implementation trial interrogation would whole product delivered other proportion. Next version would then she should have make possible at least fundamental searching.

SEZNAM POUŽITÉ LITERATURY

- [1] *Wikipedia - eLearning* [online]. 2005.[cit. 2007-05-02]. Dostupný z WWW:
<<http://cs.wikipedia.org/wiki/ELearning>>
- [2] *e-Learn: Co je to e-Learning* [online]. 2006.[cit. 2007-05-02]. Dostupný z WWW:
<http://www.e-learn.cz/uvod_coje.asp?menu=elearning&submenu=coje>
- [3] *E-learn - Blended Learning* [online]. 2007.[cit. 2007-05-02]. Dostupný z WWW:
<http://www.e-learn.cz/uvod_coje_blended.asp?menu=elearning&submenu=coje&subsubmenu=blended>
- [4] *E-learn - Klasická výuka* [online]. 2006.[cit. 2007-05-02]. Dostupný z WWW:
<http://www.e-learn.cz/uvod_coje_klasika.asp?menu=elearning&submenu=coje&subsubmenu=klasika>
- [5] *E-learn - Součásti* [online]. 2006.[cit. 2007-05-02]. Dostupný z WWW:
<http://www.e-learn.cz/uvod_soucasti.asp?menu=elearning&submenu=soucasti>
- [6] *AICC* [online]. 2007.[cit. 2007-05-02]. Dostupný z WWW: <www.aicc.org>
- [7] *Adlnet.org* [online]. 2007.[cit. 2007-05-02]. Dostupný z WWW: <www.adlnet.org>
- [8] *Studium On-line* [online]. 2007.[cit. 2007-05-03]. Dostupný z WWW:
<<http://studiumonline.cz>>
- [9] *e ACADEMY* [online]. 2007.[cit. 2007-05-03]. Dostupný z WWW: <<http://www.e-academy.cz/>>
- [10] *DOSLI* [online]. 2007.[cit. 2007-05-03]. Dostupný z WWW: <<http://web.dosli.cz>>
- [11] *Autorský zákon* [online]. 2005.[cit. 2007-05-04]. Dostupný z WWW:
<<http://business.center.cz/business/pravo/zakony/autorsky/>>
- [12] *LangMaster cz* [online]. 2007.[cit. 2007-05-04]. Dostupný z WWW:
<www.langmaster.cz>
- [13] *BlueBoard.cz* [online]. 2007.[cit. 2007-05-05]. Dostupný z WWW:
<www.blueboard.cz>
- [14] *FlashRelief* [online]. 2007.[cit. 2007-05-06]. Dostupný z WWW:
<www.flashrelief.com>

- [15] FOTR Jiří *Macromedia FLASH MX - Podrobná příručka*. 1 vyd. Praha: Computer Press, 2005. 544 s. ISBN 80-7226-677-2.
- [16] *Tiscali Foto* [online]. 2007.[cit. 2007-05-08]. Dostupný z WWW: <http://foto.tiscali.cz/mult/mult_center_030131.564559.html>
- [17] DEHAAN Jen *Flash MX 2004 Oficiální výukový kurz*. 1 vyd. Praha: SoftPress, 2004. 544 s. ISBN 80-86497-64-X.
- [18] *Sorenson Media* [online]. 2007.[cit. 2007-05-10]. Dostupný z WWW: <<http://www.sorensontech.com/>>
- [19] FRANKLIN, D., MAKAR, J. *Flash MX 2004 Actionscript - oficiální výukový kurz*. 1. vyd. : SoftPress, 2005. 904 s. ISBN 80-86497-75-5.
- [20] *VirtualDub MPEG-2 1.6.15* [online]. 2007.[cit. 2007-05-11]. Dostupný na WWW: <<http://fcchandler.home.comcast.net/stable/>>
- [21] Žaloudek Jiří *Webové služby* [online]. 2005[cit. 2007-05-13]. Dostupný z WWW: <<http://www.volny.cz/zaloudekjiri/rp/ch03.html>>
- [22] Kosek Jiří *Inteligentní podpora navigace na WWW s využitím XML* [online]. 2005.[cit. 2007-05-13]. Dostupný z WWW: <<http://www.kosek.cz/diplomka/html/index.html>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AS	Action Script
BTN	Button – Symbol Tlačítko
FPS	Frames Per Second – snímky za sekundu
GPH	Graphic – symbol Grafika
HTML	HyperText Markup Language
HW	Hardware
JS	JavaScript
LCMS	Learning Content Management System – systém pro tvorbu výukových celků
LMB	Left Mouse Button – levá tlačítka myši
LMS	Learning Management System – systém pro řízení, distribuci a vyhodnocování studia
MB	MegaByte - 2^{20} (= 1 048 576) bajtů
MC	MovieClip – Symbol Animace
MT	Motion Tween
OS	Operační systémy
PC	Personal Computer
PJ	Programovací jazyk
PT	PoinT – typografický bod
RMB	Right Mouse Button – pravé tlačítka myši
SW	SoftWare
TS	Tween Shape
WSDL	Web Services Description Language – popis obsahu webové služby
XHTML	eXtented HTML – nejnovější generace HTML jazyka
XML	eXtensible Markup Language – značkovací jazyk

SEZNAM OBRÁZKŮ

Obr. 1: PC sestava (MiniPC). Zdroj: www.GSM4YOU.cz.....	16
Obr. 2: Datový projektor. Zdroj: www.christiedigital.com.....	18
Obr. 3: Stylusy pro interaktivní tabule. Zdroj: www.AVmedia.cz.....	19
Obr. 4: Interaktivní tabule s přední projekcí. Zdroj: AVmedia.....	19
Obr. 5: Interaktivní tabule se zpětnou projekcí. Zdroj: AVmedia.....	19
Obr. 6: E-beam. Zdroj:E-beam.com.....	20
Obr. 7: Plazmová interkativní plocha. Zdroj: Image.cz.....	20
Obr. 8: Pracovní plocha.....	25
Obr. 9: Paletka Barvy.....	27
Obr. 10: Paletka zarovnání.....	27
Obr. 11: Paletka Info.....	27
Obr. 12: Detail časové osy.....	28
Obr. 13: Velikosti snímků.....	28
Obr. 14: Panel vlastností.....	30
Obr. 15: Bitmapa -100%.....	33
Obr. 16: Bitmapa - 400%.....	33
Obr. 17: Bitmapa - 650%.....	33
Obr. 18: Vektor - 100%.....	33
Obr. 19: Vektor - 400%.....	33
Obr. 20: Typy objektů (zleva): s obrysem a výplní, s ohraničením bez výplně, pouze výplní a jako poslední je objekt s ohraničením.....	34
Obr. 21: Souřadný systém.....	34
Obr. 22: Souřadnice a velikost objektu - Panel vlastností.....	35
Obr. 23: Nástroje čar: Červená - Přímky, Zelená - Křivky, Modrá - Tužka, Fialová - Štětec.. 35	
Obr. 24: Vlastnosti čar - nástroj přímka.....	36
Obr. 25: Změna koncového bodu čáry (kurzor.....	36
Obr. 26: Změna zakřivení přímky: tvar kurzoru, výsledek.....	36
Obr. 27: Křivka - nastavení směrového vektoru.....	37
Obr. 28: Editace křivky nástrojem podvýběr.....	37
Obr. 29: Tužka - i takový obrazec stačí pro doplnění na čistý obdélník.....	37
Obr. 30: Styl čáry - nástroj tužka.....	37
Obr. 31: Nástroje zaoblení a zostření čar.....	38

Obr. 32: Módy nástroje štětec.....	38
Obr. 33: Štětec - vytváří čáry s výplní.....	38
Obr. 34: Nástroj Obdélník.....	39
Obr. 35: Změna barvy výplně.....	39
Obr. 36: Vzhled kurzoru při natočení.....	40
Obr. 37: Paletka transformace.....	41
Obr. 38: Tvar kurzoru nad vrcholem.....	41
Obr. 39: Tvar kurzoru nad úsečkou.....	41
Obr. 40: Nástroj polygon (zleva): polygon; star(start point size) 0,2 a 0,7.....	41
Obr. 41: Vlastnosti textu.....	42
Obr. 42: Výběr písma včetně okna vzorku.....	43
Obr. 43: Vkládání vybraných znaků - dialog Embedding.....	44
Obr. 44: Různé orientace textu.....	45
Obr. 45: Vzdálenosti písmen (zleva):běžná, zvýšená, snížená.....	45
Obr. 46: Pozice písmen (zleva): horní index, normální, dolní index.....	45
Obr. 47: Porovnání písma s antialiasingem (nahore) a bez něj.....	46
Obr. 48: Editace symbolu na místě.....	48
Obr. 49: Orientace ve vnořených symbolech.....	48
Obr. 50: Dialog konverze na symbol (základní možnosti).....	49
Obr. 51: Klíčové snímky symbolu Tlačítko.....	51
Obr. 52: Ukázky předhotovených tlačítek.....	51
Obr. 53: Možnosti přehrávání.....	52
Obr. 54: Nastavení efektů.....	53
Obr. 55: Efekty symbol, zleva: původní symbol, pak efekty jas, zbarvení, průhlednost, pokročilé nastavení.....	53
Obr. 56: Knihovna.....	54
Obr. 57: Nastavení komprese bitmapového obrázku (musí být v knihovně symbolů).....	56
Obr. 58: Bitmapa - bez úprav.....	56
Obr. 59: Bitmapa - trace 3.....	56
Obr. 60: Shape tween se změnou polohy (z obdélníku do kružnice).....	58
Obr. 61: Indikace Shape tween.....	58
Obr. 62: Indikace Motion tweenu.....	59
Obr. 63: Vrstva vedoucí a vedená.....	59
Obr. 64: Motion tween po křivce.....	59

Obr. 65: Nastavení publikace.....	60
Obr. 66: As - Expert mod.....	62
Obr. 67: AS - Normal mod.....	63
Obr. 68: Grafický návrh č. 1.....	66
Obr. 69: Grafický návrh č.2.....	67
Obr. 70: Grafický návrh č.3.....	67
Obr. 71: Grafický návrh č.4.....	68
Obr. 72: Grafický návrh č.5.....	68
Obr. 73: Scéna prezentace, nahoře hlavička, pod ní skupina záložek.....	68
Obr. 74: Záložka.....	69
Obr. 75: Duplikování symbolu.....	70
Obr. 76: Vodící čáry.....	70
Obr. 77: Příprava na zakulacení vnitřních rohů.....	71
Obr. 78: Hotové vnitřní zakulacení.....	71
Obr. 79: Zakulacení různobarevnými obdélníky.....	71
Obr. 80: Tvar kurzoru nad tlačítkem.....	72
Obr. 81: Pojmenované klíčové snímky.....	72
Obr. 82: As - Asistent.....	74
Obr. 83: AS - Nastavení atributů pomocí asistenta.....	74
Obr. 84: Vlastnosti Loaderu.....	76
Obr. 85: Rodičovské a dceřinné obrazovky.....	76
Obr. 86: Seznam knihoven.....	77
Obr. 87: Volná transformace.....	77
Obr. 88: Přejít na konkrétní obrazovku.....	78
Obr. 89: Chování - Další obrazovka.....	80
Obr. 90: Velikost a umístění prvku.....	81
Obr. 91: Texts dynamickou šířkou.....	82
Obr. 92: Text s pevnou šířkou.....	82
Obr. 93: Příprava nože.....	82
Obr. 94: Obrýsy nože.....	83
Obr. 95: Dokončený závitový nůž.....	83
Obr. 96: Vřeteno a upínací deska.....	83
Obr. 97: Ovládací panel a dělník.....	83
Obr. 98: Kompletní obrázek čelního soustruhu.....	84

Obr. 99: Animace tangenciálního nože.....	85
Obr. 100: Tangenciální nůž.....	86
Obr. 101: Tříška před obrobkem.....	86
Obr. 102: Časová osa animace.....	86
Obr. 103: Klíčové snímky neviditelného tlačítka.....	88
Obr. 104: Indikace zvuku ve snímku.....	88
Obr. 105: Textová oblast (TextArea).....	90
Obr. 106: Popis (Label).....	90
Obr. 107: Tlačítko (Button).....	90
Obr. 108: Vstupní text (TextInput).....	90
Obr. 109: Roletková nabídka (Combobox).....	90
Obr. 110: Číselník (NumericStepper).....	90
Obr. 111: Zaškrtačací políčko (CheckBox).....	90
Obr. 112: Paletka webových služeb, zobrazeny dostupné služby registru.....	90
Obr. 113: Přidání chování - odeslat data.....	91

SEZNAM TABULEK

Tab. 1: Cena minimální konfigurace PC k 1.5.2007 (vše bez DPH).....	16
Tab. 2: Rozlišení dataprojektorů.....	17
Tab. 3: Kalkulace finančních nákladů (ceny jsou uvedeny bez DPH).....	23