

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA PEDAGOGICKÁ

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**REALIZACE ELEKTRONICKÉHO VÝUKOVÉHO KURZU
SE ZAMĚŘENÍM NA TVORBU PLOŠNÝCH VEKTOROVÝCH
ANIMACÍ**

DIPLOMOVÁ PRÁCE

Tereza Příbylová

*Učitelství pro základní školy, obor Informatika - Matematika
léta studia (2014 - 2016)*

Vedoucí práce: Mgr. Denis Mainz

Plzeň, 2016

Prohlašuji, že jsem diplomovou práci vypracovala samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 30. června 2016

.....
vlastnoruční podpis

Poděkování

Touto cestou bych ráda poděkovala vedoucímu mé diplomové práce, panu Mgr. Denisu Mainzovi, za vstřícný přístup, užitečné rady a připomínky při zpracování mé diplomové práce. Dále bych ráda poděkovala své rodině a přátelům, kteří mi byli během celé doby studia velkou oporou.

ZDE SE NACHÁZÍ ORIGINAL ZADÁNÍ DIPLOMOVÉ PRÁCE
A HNED ZA NÍM
ROZHODNUTÍ.

OBSAH

SEZNAM ZKRATEK	2
ÚVOD.....	3
1 ANIMACE	4
1.1 DEFINICE.....	4
1.2 HISTORIE A VÝVOJ ANIMACE.....	4
1.3 PRINCIP ANIMACE	8
1.3.1 Iluze pohybu	8
1.3.2 Princip.....	9
1.4 POČÍTAČOVÁ ANIMACE	10
1.4.1 2D a 3D	10
1.4.2 Terminologie.....	10
1.5 FORMÁTY DVOUROZMĚRNÉ VEKTOROVÉ ANIMACE	11
1.5.1 Formát SWF	11
1.5.2 SVG	12
1.5.3 Jazyk SMIL	14
1.5.4 Jazyk VML.....	15
1.5.5 Jazyk XML	15
1.6 PROGRAMOVÉ NÁSTROJE PRO TVORBU DVOUROZMĚRNÝCH VEKTOROVÝCH ANIMACÍ.....	15
1.6.1 Přehled.....	15
1.6.2 Popis vybraných programových nástrojů.....	16
1.6.3 Výběr programových nástrojů k tvorbě plošných vektorových animací využitých ve výukovém kurzu	17
2 PŘÍPRAVNÁ FÁZE TVORBY ELEKTRONICKÉHO VÝUKOVÉHO KURZU	19
2.1 FÁZE PŘÍPRAVY REALIZACE VÝUKOVÉHO KURZU	19
2.2 NÁVRH USPOŘÁDÁNÍ VÝUKOVÉHO KURZU	19
2.3 TVORBA STUDIJNÍCH ČLÁNKŮ	20
2.3.1 Animace	20
2.3.2 SVG animace	21
2.3.3 Animace v editoru.....	44
3 VÝBĚR VHODNÉHO SOFTWARE PRO TVORU ELEKTRONICKÉHO VÝUKOVÉHO KURZU.....	67
3.1 AUTORSKÝ SYSTÉM PROAUTHOR	67
3.2 CAMTASIA STUDIO 8	68
3.3 INKSCAPE	68
4 REALIZACE ELEKTRONICKÉHO VÝUKOVÉHO KURZU	69
4.1 CÍL ELEKTRONICKÉHO VÝUKOVÉHO KURZU	69
4.2 ELEKTRONICKÝ VÝUKOVÝ KURZ	70
ZÁVĚR	72
RESUMÉ.....	73
SEZNAM LITERATURY	74
SEZNAM OBRÁZKŮ	76

SEZNAM ZKRATEK

CSS	kaskádový styl (jazyk pro vytvoření stylu webových stránek)
GIF	grafický formát rastrové grafiky (bezeztrátová komprese)
SMIL	vektorový formát využívající syntaxi XML a jeho největší výhoda je v tom, že umožňuje vytváření multimedálních prezentací
SVG	značkovací jazyk a formát souboru, který popisuje 2D grafiku pomocí jazyka XML
SWF	nejznámější formát zobrazování vektorově založených animací, multimédií a interaktivního obsahu na webu
XML	značkovací jazyk pro vytváření aplikací, výměnu dat a publikování dokumentů
fps	frame per second = snímků za sekundu
GUI	grafické uživatelské rozhraní

ÚVOD

Hlavním cílem této diplomové práce je vytvořit elektronický výukový kurz, který jeho uživatelům umožní proniknout do problematiky tvorby plošné vektorové animace a seznámí je s principy a základními možnostmi její tvorby.

První kapitola seznamuje s definicí animace, jejími principy a terminologií, dále pak obsahuje přehled nejznámějších formátů dvourozměrné vektorové animace, přehled vybraných programových nástrojů pro tvorbu animací a výběr programového nástroje, který bude součástí kurzu.

V druhé kapitole je popsána tvorba náplně výukového kurzu. Jsou zde uvedeny stanovené fáze tvorby elektronického kurzu, jeho struktura a náplň jednotlivých studijních článků doplněna o názorné ukázky.

Ve třetí kapitole jsou popsány programy, pomocí kterých byl výukový kurz vytvářen.

Závěr práce pojednává o samotné realizaci výukového kurzu. Jsou zde uvedeny cíle kurzu a stručný popis náplně doplněný o ukázky hotového kurzu.

Animace budou vytvořeny v programu Macromedia Captivate, který umožňuje snímat práci v programu a doplnit ji o popisky aktuálního dění.

1 ANIMACE

Když se řekne animace, jako první si většina lidí vybaví spojitost s animovanými filmy. V dnešní době se ale animace nevyužívá jen pro tvorbu animovaných filmů, ale také v oblasti vědy, v počítačových hrách, v reklamách (ať už v televizi nebo na internetu), divadle, kultuře a v mnoha dalších oblastech. Animace má velký potenciál a obrovskou schopnost změnit se v manipulační nástroj. Při sledování animace se přesouváme do uměle vytvořeného světa, do kterého se vžíváme, i když jde pouze o imaginární svět animace. Např. dobře je tato schopnost vidět u počítačových her, kde se hráč vžívá do postavy, za kterou hraje. Můžeme tak vytvořit i animaci něčeho, co ve skutečnosti neexistuje nebo je těžké provést, např. pokusy, simulace atp. V dnešní době také animace nevytváří už jen lidé, ale k tvorbě používají pomoc počítače.

1.1 DEFINICE

Pojem animace pochází z cizího slova znamenajícího ožívování, oduševnění, vytváření zdánlivě se pohybujících věcí. Do češtiny se toto slovo dostalo z Francie (animation) a z italtštiny (animazione). Samotné slovo animace, které zdomácnělo v romanských jazycích, ve svém původním významu vychází z latinského výrazu „anima“, česky duše. Animování tedy značí „odduševnění“, kde se slovo „duše“ chápe jako ekvivalent slova „život“. Proto slovo „animace“ chápeme jako oživení, naplnění životem nebo duchem. [5]

1.2 HISTORIE A VÝVOJ ANIMACE

První výskyt animací je znám z kreslených filmů, kde se jednotlivé scény vykreslovaly tak, aby vytvářely iluzi pohybu. V té době bylo potřeba vykreslit nejméně 24 obrázků za vteřinu. V současné době je pod pojmem animace zahrnuta celá škála činností.

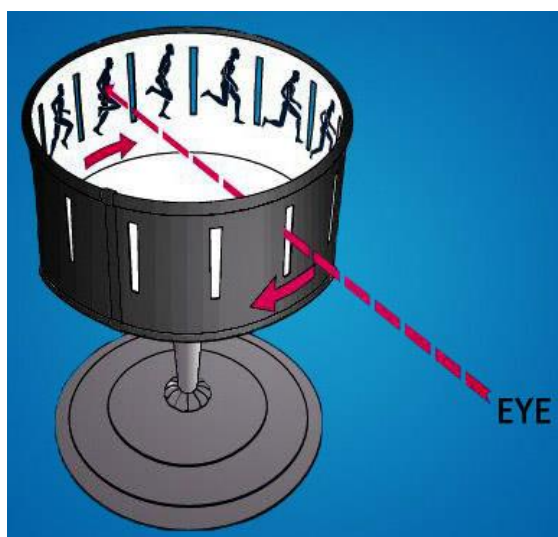
- První pokusy animace jsou známé už z doby kamenné, kdy v jeskyních kreslili zvířata v jednotlivých fázích pohybu za sebou a tím naši předkové znázorňovali pohyb. [10]
- První opravdová zařízení, která demonstrovala iluzi pohybu, vznikly v 19. století, kdy belgický fyzik Joseph Antoine Ferdinand Plateau v roce 1832 představil zařízení nazvané phenakistoscope. Šlo o disk, kde kolem středu bylo nakresleno několik obrázků v různých fázích pohybu. Rotací docházelo k iluzi pohybu. [10]



Obrázek 1: Phenakistoscope

Zdroj: *Phenakistoscope - England - 1833* [online]. [cit. 2016-04-20]. Dostupný z: <http://dickbalzer.tumblr.com/page/7>

- V roce 1834 britský matematik William George Horner představil technologicky vyspělejší zařízení, než byl phenakistoscope a to zoetrope. Bylo to první zařízení tohoto typu, kdy animaci mohlo sledovat více lidí současně. Šlo o válec s vyříznutými otvory po stranách, do kterých divák po roztočení koukal, a tím docházelo k iluzi pohybu. [10]



Obrázek 2: Zoetrope

ZOETROPE. *Parlor Toy Project* [online]. [cit. 2016-04-20]. Dostupný z: http://zoetrope4.weebly.com/uploads/1/6/3/0/16309614/3857102_orig.jpg?236

- V roce 1877 představil francouzský vynálezce Charles-Émile Reynaud přístroj podobný zoetrope, avšak místo otvorů po stranách se uprostřed válce nacházela

zrcadla, která odrážela obrázky od bočních stěn. Tento přístroj se jmenoval praxinoscope. [10]

- K největšímu rozmachu animace došlo až s rozmachem kinematografie. [10]
- Za otce počítačové animace je považován americký animátor John Whitney. Ve spolupráci se svým bratrem Jamesem vytvořil po druhé světové válce za pomoci analogového počítače (spirografu) vytvořeného z protiletectvých počítačů z druhé světové války, sérii experimentálních filmů. [12]
- V roce 1960, grafik William Fetter zavedl pojem „počítačová grafika“ pro popis nového způsobu designu. [12]
- V roce 1962, student doktorského studia na Massachusettském technologickém institutu, Ivan Sutherland, pracoval na své disertační práci. V té se zabýval vývojem systému pro grafickou komunikaci mezi člověkem a počítačem. Součástí práce byl první GUI program pro interaktivní práci s počítačovou animací, Sketchpad. Šlo o první interaktivní kreslicí program, který umožňoval manipulovat s obrazem na monitoru v reálném čase a považuje se za předchůdce dnešních grafických CAD systémů. Jako první také Sutherland v tomto programu využil pro popis geometrických útvarů principy objektově orientovaného programování a vymyslel pop-up menu. [12]
- V roce 1963 vznikl první animovaný film, který simuloval oběh družice okolo Země. Vytvořil ji Edward Zajac ve výzkumné vědecké společnosti Bell Labs. [12]
- V roce 1965 na Univerzitě v Utahu, založil David Evans fakultu výpočetní techniky, kde se začali vyvíjet základní techniky trojrozměrné počítačové grafiky. [12]
- V roce 1968 Sutherland spolu s Evansem založili společnost Evans & Sutherland, která se zabývala výzkumem v oblasti provádění výpočtu pro počítačovou grafiku v reálném čase a vývojem hardwarových prostředků, které toto měli umožňovat. [12]
- V roce 1968 skupina sovětských matematiků a fyziků spolu s Nikolajem Konstantinovem vytvořili matematický model demonstrující pohyb kočky. Vymysleli program pro řešení diferenciálních rovnic, který tento pohyb popisoval.

Po vytištění stovek snímků s použitím znaků abecedy, vznikla animace postavy chodící kočky. [12]

- Od roku 1960 vznikl a vyvíjel se systém Scanimate, na kterém se podílel Lee Harison. Vznikl z prehistorického systému pro technologický záznam pohybu a umožňoval vytváření animací v reálném čase. Od roku 1967 se tento systém velmi rozvinul a ovlivnil grafiku televizního obrazu i filmu v celých 80. letech. V té době se také v souvislosti s počítačově zpracovaným obrazem poprvé objevily pojmy „key“ a „layer“. [12]
- V roce 1969 světové centrum pro umění animace v Kanadě začíná s experimenty v oblasti výpočetní techniky a umělec Peter Foldes vytvořil v roce 1971 film, ve kterém využil animační techniku známou jako interploaci (inbetweening, morphing), která spočívá v postupné animaci od jednoho obrázku k druhému. [12]
- V roce 1972 v hlavním centru počítačové animace v Británii, animátor, grafický designér a softwarový vývojář Alan Kitching vyvinul 2D vektorový animační software Antics, který umožňoval uživatelům vytvářet animace bez nutnosti programování. Nejdříve běžel pod systémem Unix, poté pod systémy založené na C++, nakonec i pro Windows a Apple MAC. [12]
- V roce 1974 vznikla velmi významná síť organizující zájemce o počítačovou grafiku a interaktivní technologie – SIGGRAPH. Tyto konference se konají dodnes a prezentují se na nich současné i budoucí technologie, umělecká díla vytvořená počítači atp. [12]
- 70. léta byla nejdůležitější pro vývoj počítačové animace. Rozvíjela se trojrozměrná grafika, objevily se první základní renderovací techniky a vznikaly první animované filmy. V roce 1969 je první zmínka o využití framebufferu¹ a v roce 1972, Richard Shoup vytvořil program SuperPaint, který framebuffer používal. [12]

¹ framebuffer = snímková vyrovnávací paměť (paměť, kam se ukládá obraz před odesláním na obrazovku)

- V 80. letech nastal velký rozmach ve vývoji komerčního hardwaru, zejména v začlenění framebufferové technologie a v cenové dostupnosti. Byly vyvinuty animace pro TV seriál, které popisovali fyzikální zákony a matematiku. Také v kombinaci s ruční animací byly počítačově dotvořeny první filmy od Walta Disneyho. [12]
- V 90. letech počítačová animace expandovala ve filmu a v televizi. Došlo opět ke zlepšení hardwarových prostředků, cenové dostupnosti a stále rostoucí nabídce softwarových nástrojů pro tvorbu počítačové animace. V roce 1995 vznikl první plně počítačem animovaný celovečerní film od Disney a Pixarů – Toy Story. Začala se také využívat technologie Motion capture² a Matchmoving³. S rozvojem internetu přišla společnost Macromedia na trh s grafickým vektorovým programem Flash, kterou v roce 2005 odkoupila firma Adobe. Na konci 90. let také začal vývoj vektorového formátu SVG. [12] [11]
- Od konce 90. let až do současnosti je vývoj 2D i 3D animací ve velkém rozmachu. Dochází neustále k vylepšení technologií, rozvoje softwaru a tím k obrovským možnostem tvorby animací. Filmy se začaly natáčet na digitální video a vzrostl zájem o počítače a platformové hry. 2D animace má za sebou dlouhou cestu. Přes nespočet animací přenesla diváky do světů fantazie a to přes její dovednost představivosti. Nejdříve to byly jen krátké animace, které byly časově náročné a dnes díky počítačové technice a softwaru jsou animace přiváděny k čím dál větší dokonalosti. A i když 3D animace jsou dnes velmi populární, 2D animace budou mít ve světě pořád své místo. [12]

1.3 PRINCIP ANIMACE

1.3.1 ILUZE POHYBU

Vnímáme svět pomocí svých smyslů. V tomto případě jde o naše oči. Náš mozek pak všechny informace, které vidíme, vyhodnotí. Velkou část našeho vnímání tvoří právě obrazové informace. Jak dva rozdílní lidé vidí jednu a tu samou informaci už závisí na jejich osobnosti, úhlu pohledu a jejich představivosti, protože vidění je psychickou funkcí

² Motion Capture neboli Mocap = snímání pohybu a jeho následné převedení do digitálního modelu.

³ Matchmoving = proces dosazování počítačem vytvořené grafiky do reálného záběru

a ta může být ovlivněna různými faktory. Proto to není jen o tom, co jen vidíme, ale přidávají se k tomu i naše zkušenosti a fantazie. V praxi je toto pravidlo označeno jako zákon pregnantnosti⁴. A jakou souvislost to má s animací?

Animace je ve své podstatě jen iluzí pohybu. Skutečnost je taková, že člověk vnímá pohyb, ale jde jen o klamné zdání. Všechno je to v naší hlavě, konkrétně v nedokonalosti našich očí. Fyziologie našeho oka způsobuje, že po sobě jdoucí snímky animace vnímáme jako ucelený pohyb. Obraz zachycený na sítnici našeho oka zde zůstane po dobu několik setin vteřiny, než se sítnice opět zregeneruje. Tomuto fyziologickému jevu se říká setrvačnost lidského oka⁵. Jde o to, že pokud pozorujeme upřeně nějaký objekt a potom se podíváme jinam nebo tento objekt zmizí z našich očí, na desetinu sekundy se nám obraz uchová na sítnici našich očí. Dochází k tomu, že se v našich očích překrývají dva obrazy (ten, na který jsme se dívali před desetinou sekundy a ten, na který se díváme právě teď). Překrývání nám umožní iluzi plynulého pohybu namísto několika oddělených statických obrazů, které jeden po druhém vytváří výsledný pohyb. Pokud by naše oči touto vlastností nedisponovaly, nejen, že bychom nemohli sledovat filmy a televizi, ale nebyli bychom schopni pozorovat jakýkoli pohyb v našem okolí. [6]

1.3.2 PRINCIP

K iluzi pohybu stačí člověku počáteční a konečná fáze pohybu, pak je na řadě náš mozek a naše psychika, která nám poskytne výsledný pohyb. Iluze pohybu tedy vzniká díky dostatečně rychlému promítání jednotlivých statických obrázků, které pak díky setrvačnosti lidského oka člověk vnímá jako plynulý pohyb. Počáteční a koncový bod pohybu objektu tvoří klíčové snímky. Mezi nimi se objekt postupně přemísťuje a výsledkem jsou mezisnímky. Čím více mezisnímků mezi klíčovými snímky je, tím je lidské oko nedokáže postřehnout a iluze pohybu je plynulejší. Minimální hodnota pro plynulou animaci je 24 snímků za sekundu (fps). Nejčastěji se používají hodnoty 24, 25, 30 nebo 60 fps. [6]

⁴ Celek není jen souhrnem částí, vytváří tzv. Gestalt (tvar). Zdroj: [POGRANOVÁ, Anna. Vnímání. *Artslexikon* [online]. [cit. 2016-04-29]. Dostupné z: <http://www.artslexikon.cz//index.php?title=Vn%C3%ADm%C3%A1n%C3%AD>

⁵ Roku 1824 tento jev definoval anglický učenec Peter Mark Roget ve svém díle O zachování obrazu pohybujících se předmětů. Zdroj: Prehistorie filmu [online]. 2011 [cit. 2016-05-29]. Dostupné z: http://nemy-film.sweb.cz/Prehistorie_filmu/Uvod.html

1.4 POČÍTAČOVÁ ANIMACE

Animace vytvořená jen prostřednictvím počítačové techniky. Vzniká pomocí speciálních animačních programů, kterých je v dnešní době celá řada. Může jít jak o dvourozměrnou animaci, tak o trojrozměrnou animaci.

1.4.1 2D A 3D

- 2D animace = dvourozměrná (plošná) animace – pracuje s dvourozměrnými objekty (bitmapovými obrázky, geometrickými objekty), může být bitmapová nebo vektorová.
- 3D animace = trojrozměrná animace – nejznámějším využitím je vytváření animací pro tvorbu filmů, počítačových her a dále ve vědě a průmyslu (počítačové simulace, modely objektů atp.).

1.4.2 TERMINOLOGIE

- frame = snímek – animace se skládá z mnoha jednotlivých obrázků (snímků), ty bývají statické a v každé sekundě animace proběhne několik snímků těchto statických obrázků
- fps = frame per second – počet snímků za sekundu
- keyframing = klíčování
- layer = vrstva
- timeline = časová osa
- klíčový snímek (keyframe) = snímky, které se zobrazí na počátku animace a na konci animace objektu (např. první klíčový snímek bude stojící postava člověka a druhý klíčový snímek bude sedící postava člověka). [1]
- mezisnímky = všechny snímky mezi dvěma klíčovými snímky. Animační editory si mezisnímky automaticky dopočítávají a vygenerují (tento proces se nazývá inbetweening nebo tweening). [1]

1.5 FORMÁTY DVOUROZMĚRNÉ VEKTOROVÉ ANIMACE

Grafický formát je určitým standardem popisujícím způsob, jakým jsou grafická data ukládána a každý formát má své vlastnosti, které ho charakterizují. Grafických formátů existuje obrovské množství a bývá pravidlem, že každá společnost vyvíjející grafické software, má svůj nativní firemní formát. Grafické formáty můžeme dělit na vektorové a rastrové a dále podle komprese dat na ztrátové a bezztrátové. V této podkapitole se zaměřím na nejpoužívanější formáty, které mají vztah k problematice dvourozměrné vektorové animace.

1.5.1 FORMÁT SWF [7]

Zkratka SWF značí *Small Web Format* (dříve *Shockwave Flash*). Jde o kontejnerový formát s příponou *.swf* vyvinutý společností FutureWave v polovině roku 1990. Od roku 1996 byl ve vlastnictví společnosti Macromedia, která společnost FutureWave odkoupila a od konce roku 2005 je ale ve vlastnictví společnosti Adobe a je nejnámějším formátem zobrazování vektorově založených animací, multimédií a interaktivního obsahu na webu. Účelem vytvoření takového formátu bylo jednodušší zavedení vektorové grafiky a animací na webu (reklamní bannery, hry, videa apod.). Účel vytvoření tohoto formátu byla jeho velikost po komprimaci a možnost zasílání a zobrazování přes pomalejší síť. Formát podporuje možnost animace, kombinace s multimédií a jeho největší předností je možnost interaktivity. Nativně lze tento formát získat ze softwaru ve vlastnictví Adobe (konkrétně Adobe Flash), ale existují i open source nástroje, ve kterých můžeme SWF soubor editovat nebo další programy, které nabízejí možnost exportu do tohoto formátu. Jako ostatní soubory, které jsou produkovány společností Adobe, je potřeba mít pro přehrání SWF nainstalován přehrávač Adobe Flash Player, který je volně dostupný a má ho dnes jako doplněk webového prohlížeče většina z nás.

Výhody:

- možnost animace, přidání multimédií,
- interaktivita – pomocí jazyka ActionScript,

- podpora anti-aliasingu⁶,
- rychlé a kvalitní vykreslování v libovolném barevném formátu,
- zpětná kompatibilita – vývoj nových funkcí a zachování kompatibility i se staršími verzemi Flash Playeru,
- zobrazení fontu písma bez závislosti na tom, zda je font v daném zařízení nainstalován,
- binární formát – nelze do něj nahlédnout jako např. do SVG.

Nevýhody:

- tvorba v produktech Adobe,
- může být náročnější na výkon počítače.

1.5.2 SVG [8]

Zkratka SVG z anglického Scalable Vector Graphics znamená v překladu *Škálovatelná vektorová grafika*.

- Scalable = možnost libovolného zvětšování a zmenšování bez ztráty kvality
- Vector Graphics = matematicky definované geometrické objekty, ze kterých jsou SVG objekty tvořené

Jde o značkovací jazyk a formát souboru popisující dvourozměrnou vektorovou grafiku prostřednictvím značkovacího jazyka XML. Zároveň je otevřeným standardem vyvinutým konsorciem W3C roku 1999 a pro vytvoření nebo úpravu souboru nám postačí obyčejný textový editor. Samozřejmě existuje i velké množství editorů, které s formátem SVG umí pracovat (jde o komerční produkty ale i o volně dostupné „open source“ programy). SVG ale neumožňuje jen ukládání vektorové grafiky, ale i propojení s rastrovou grafikou, vložení textu a umožňuje i tvorbu animací a interaktivitu. Proto SVG není jen nástrojem k zobrazení statických obrázků (loga, ikony, prvky uživatelského rozhraní, ilustrace), často se používá pro interaktivní mapové portály (GIS), jednodušší hry, online grafické editory, interaktivní grafické prvky apod.

⁶ Anti-aliasing = vyhlazení hran a tím vytvoření dojmu vyššího rozlišení snímku

Objekt vytvořený v SVG je složen ze základních geometrických tvarů (obdélník, kruh, elipsa, čára, lomená čára, mnohoúhelník), cest, textu (není tvořen cestami), podporuje i možnost ořezu, průhlednosti a práce s rastrovou grafikou. Jednotlivé objekty mohou být seskupovány, formátovány pomocí atributů nebo CSS apod. Objekty můžeme měnit pomocí knihovny DOM, použít CSS styly nebo jazyk JavaScript.

Zdrojový kód lze uložit buď jako soubor s příponou .svg, tím vznikne plnohodnotný obrázek nebo můžeme kód vložit do XHTML/HTML5 kódu.

V současnosti už většina webových prohlížečů formát SVG podporuje bez nutnosti instalace zásuvných modulů. Jen u verzí Internet Explorer 8 a nižší a mobilních operačních systémů Android 2.3 a nižší.

Výhody:

- libovolné zvětšování a zmenšování bez ztráty kvality a změny datového objemu,
- možnost vložení obrázků přímo do webové stránky,
- možnost úpravy vzhledu pomocí CSS,
- možnost manipulace s objekty pomocí JavaScriptu,
- možnost animování jednotlivých prvků a efektů SVG obrázku,
- možnost vložení textu (skutečného, ne cest, které text utváří) a tím i možnost vyhledávání podle obsaženého textu.

Nevýhody:

- podpora ve webových prohlížečích,
- vhodnost pro všechny typy grafik,
- datový objem může být větší než u optimalizovaných rastrových formátů,
- otevřený zdrojový kód (ochrana před kopírováním a upravováním).

1.5.2.1 Verze

- SVG 1.0 – v září 2001 se stal standardem konsorcia W3C.

- SVG 1.1 – v lednu 2003 vydáno rozšíření, které přidalo formátu větší možnosti využití. Byla to možnost spolupráce s CSS styly, přidání přímého propojení SVG s ostatními standardy používanými na webových stránkách a to např. propojení s HTML5, XHTML a MathML (pro účely vědecké komunikace), SMIL (pro účely vytváření animací) a s GML (pro geografické využití).
- SVG Tiny 1.1 a SVG Basic – verze, které vychází ze SVG 1.1 a jsou určeny pro mobilní telefony a PDA pro svou menší náročnost).
- SVG Tiny 1.2 – prosinec 2008.
- SVG 1.1 Second Edition – srpen 2011.
- SVG 2 – právě vyvíjená verze.

1.5.3 JAZYK SMIL [13]

Zkratka SMIL (z anglického Synchronized Multimedia Integration Language) znamená v překladu *Jazyk pro integraci a synchronizaci multimediálního obsahu*. Jde o vektorový formát využívající syntaxi XML, který je standardem konsorcia W3C. Jeho největší výhodou je v tom, že umožňuje vytváření multimediálních prezentací, které mohou obsahovat video, hudbu, text a obrázky. Sám o sobě neslouží k vytváření obrázků nebo videí, ale umožňuje integraci multimediálních objektů do statických dokumentů HTML nebo XML. Můžeme tedy vkládat do dokumentu hotové objekty (multimédia), vymezit jim prostor a určit jim, kdy se mají zobrazit atp. Průběh je řízen podle nastaveného času nebo uživatelem (interakce). Výsledkem je, spíše než dokument XML, dynamická prezentace. Je jedním ze tří možností, jak vytvářet SVG animace.

1.5.3.1 Verze

- 1.0 – od roku 1998,
- 2.0 – od srpna 2001 (použití např. pro animace v SVG),
- 2.1 – od prosince 2005 (použití např. pro zprávy MMS),
- 3.0 – od prosince 2008.

1.5.4 JAZYK VML [9]

Jde stejně jako u SVG o vektorový značkovací jazyk k popisu vektorové grafiky (z anglického Vector Markup Language) na základě značkovacího jazyka XML. Vyvinula ho společnost Microsoft a jeho vlastnosti odpovídají vlastnostem formátu SVG. Tento formát je již dnes v podstatě nepoužívaný, protože měl podporu jen v prohlížečích Internet Explorer verze 5 a vyšší a v ostatních webových prohlížečích grafika ve formátu VML nelze zobrazit.

1.5.5 JAZYK XML [14]

Obecný „rozšiřitelný“ značkovací jazyk (z anglického Extensible Markup Language), který byl vyvinut a standardizován konsorciem W3C (udává standardy pro WWW). Je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů (popis struktury z hlediska věcného obsahu, nezabývá se vzhledem).

1.6 PROGRAMOVÉ NÁSTROJE PRO TVORBU DVOUROZMĚRNÝCH VEKTOROVÝCH ANIMACÍ

Programů a editorů na tvorbu a úpravu vektorové grafiky existuje pár desítek. Některé z nich nejsou skoro vůbec známé, některé jsou placené, jiné zase fungují jako online editory bez potřeby instalace do počítače. Pro diplomovou práci jsem si vybrala několik těchto editorů, které se pokusím porovnat.

1.6.1 PŘEHLED [16]

Pro přehled uvádím několik vektorových editorů pro tvorbu dvourozměrných animací.

1.6.1.1 Open-source licence

- Ajax Animator (webová aplikace)
- Tupi (dříve KToon)
- Pencil2D
- SWFTools
- Synfig Studio

1.6.1.2 Freeware

- Flipnote Studio
- Pivot Stickfigure Animator
- Animatron (webová aplikace)

1.6.1.3 Trialware

- Adobe After Effects
- Adobe Animate CC (dříve jako Adobe Flash Professional)
- Anime Studio
- Antics 2-D Animation
- ToonBoom
- Toonz

1.6.1.4 Komerční

- Autodesk Animator
- Cacani
- Motion (jen OS X)
- RETAS

1.6.2 POPIS VYBRANÝCH PROGRAMOVÝCH NÁSTROJŮ

Synfig Studio

Celkem oblíbený vektorový grafický editor dvourozměrných animací, volně dostupný multiplatformní editor pro ty, kteří s dvourozměrnou animací začínají. Podporuje kreslení ve vrstvách, automatické vykreslování mezisnímků mezi klíčovými snímky, importování bitmapových obrázků atd. [17]

Pencil2D

Multiplatformní, volně dostupný editor, který používá bitmapové a vektorové kreslicí rozhraní k tvorbě dvourozměrné grafiky. Primárně není určen pro tvorbu animací. [18]

ToonBoom Studio

Komplexní nástroj pro tvorbu animací, umožňující export ve vysokém rozlišení (i pro HD produkci). Je to poloprofesionální nástroj, který používají různá animační studia i profesionální animátoři. Umožňuje i simulaci trojrozměrného prostoru realizovanou pohybu kamery v prostoru kolem objektu. Je vhodný i pro amatérské animátory a fanoušky animace. Jeho cena je 1000 USD a je dostupný pro OS Windows a Mac OS. [19]

Adobe Animate CC

Nástroj pro tvorbu interaktivních SWF animací pomocí špičkových nástrojů. Patří mezi nejpoužívanější programy pro tvorbu dvourozměrných animací (primárně webových). Možnost vyzkoušet zkušební verzi nebo zakoupit roční licenci za 290 EUR. [20]

1.6.3 VÝBĚR PROGRAMOVÝCH NÁSTROJŮ K TVORBĚ PLOŠNÝCH VEKTOROVÝCH ANIMACÍ VYUŽITÝCH VE VÝUKOVÉM KURZU

Pro výukový kurz jsem si zvolila webovou aplikaci vektorového animačního editoru Animatron, který je dostupný na adrese <https://www.animatron.com/>. Po registraci a přihlášení je přístupný odkudkoliv, není potřeba instalace a můžeme ihned pracovat. A i když nenabízí prostředí v češtině, práce v něm je natolik intuitivní a prostředí uživatelsky velmi přívětivé, že není potřeba před prací studovat několikastránkové manuály. [21]

Vznik

Cílem bylo vytvořit uživatelsky přívětivý animační editor, který umožní tvorbu animovaného a interaktivního obsahu ve webovém prohlížeči. Inspirací pro vytvoření tohoto nástroje byl domácí úkol desetileté dcery spoluzakladatele a CEO Dmitrije Skavishe. Doporučené programy pro tvorbu, které měla jeho dcera ze školy, byly složité a nepřinášely rozšiřující možnosti pro tvorbu animace. i přes jeho snahu najít lepší program, došel k závěru, že žádný takový program neexistuje. A tím se zrodila myšlenka na vytvoření Animatronu. [21]

Vývoj

Ve vývoji je aplikace už od roku 2011. Vyvinula ji společnost JetBrains, která je známá jako nejlepší vývojářská společnost na světě. Nápad na vývoj Animatronu se jim líbil a tak tento

projekt zafinancovali. Dnes má společnost kolem 20 lidí, kteří neustále pracují na dalším vývoji aplikace.

Aplikace je psána v Javě a samotná má jen 800 KB. Je dostupnější než Adoba Animate CC nebo ToonBoom, ale jeho sada nástrojů a funkcí se dokáže vyrovnat s prostředím Adobe Animate CC. [21]

Přístup

V Animatronu není nutné se registrovat, můžeme v editoru pracovat i jako hosti. To ale přináší své nevýhody. Spousta možností je omezena. Nejlepší varianta je zaregistrovat se buď přes existující účet na Googlu, Facebooku, Twitteru, Dribble nebo zadat e-mailovou adresu, na kterou nám přijde potvrzující e-mail. Tím získáme účet FREE, který je bezplatný. Má však také určitá omezení.

- FREE účet – vytvoření 5 veřejných projektů, publikování do HTML5, export do SD videa, export do SVG, SVG SMIL, PNG a GIF, importování minutového SD videa, audia, 2 GB úložiště.
- PRO účet – neomezené veřejné i soukromé projekty, export bez vodoznaku, pokročilé možnosti, export do HD videa, import desetiminutového videa, 10 GB úložiště, cena: 30 USD (měsíc) nebo 180 USD (rok).
- BUSINESS účet – export Full HD Vide, prémiová podpora, 100 GB úložiště, cena: 60 USD (měsíc) nebo 360 USD (rok).

Vlastnosti editoru

- Import obrázků, audia a videa.
- Export do SVG, SVG SMIL, GIF a PNG a publikování do HTML5.
- Možnost sdílení na sociálních sítích.
- Možnost interaktivity a skriptování.
- Veřejná knihovna vytvořených animací.
- Své práci můžeme také sdílet ve veřejné knihovně Animatronu.

Spolupráce více lidí na projektu v reálním čase. [21]

2 PŘÍPRAVNÁ FÁZE TVORBY ELEKTRONICKÉHO VÝUKOVÉHO KURZU

Tato kapitola popisuje jednotlivé fáze tvorby výukového kurzu, návrh jeho uspořádání, tvorbu studijních článků a multimediálních komponent.

2.1 FÁZE PŘÍPRAVY REALIZACE VÝUKOVÉHO KURZU

V první fázi jsem si prostudovala vhodnou literaturu, ze které jsem následně začala tvořit jednotlivé studijní články.

V další fázi jsem k těmto studijním článkům připravovala vhodné multimediální komponenty (obrázky, animace a videa). Bylo nutné upravit je do takových formátů, které lze vkládat do prostředí ProAuthoru, proto jsem v této fázi zaznamenávala práci na obrazovce, pracovala v grafických editorech, stříhala video a vytvářela obrázky.

Po přípravě textových i multimediálních komponent přišlo na řadu stanovení si návrhu uspořádání kurzu a členění jednotlivých kapitol a studijních článků.

Poslední fází bylo naplnění výukového kurzu připravenými textovými a multimediálními komponentami.

2.2 NÁVRH USPOŘÁDÁNÍ VÝUKOVÉHO KURZU

Výukový kurz jsem se rozhodla rozčlenit na tři hlavní kapitoly:

- Animace
- SVG animace
- Animace v editoru

První kapitola *Animace* bude obsahovat tři studijní články. Prvním z nich bude studijní článek *Definice animace*, který pojednává o původu názvu slova animace, co pro nás znamená animace v současné době a jaká je její historie. V dalším článku *Princip animace* bude popsáno, na jakém principu animace funguje. A v posledním článku *Fáze tvorby animace* bude uveden příklad toho, jak bychom si měli rozmyslet scénář animace předtím, než ji samotnou začneme tvořit.

Druhá kapitola *SVG animace* bude obsahovat dvě části. První část bude tvořena pěti studijními články: *Formát SVG*, *Základní struktura SVG kódu*, *Společné vlastnosti elementů*,

Základní elementy, Další elementy, ve kterých se studenti kurzu seznámí s vlastnostmi samotného formátu SVG, následně na to se zápisem kódu a jeho částmi. Poté se seznámí s elementy a jejich zápisem. Zápisy budou vhodně doplněny o ukázky. Tato část bude uzavřena autotestem, kterým si studenti otestují nastudované znalosti a v další části si na úkolu budou moci vyzkoušet samotný zápis vlastního kódu. Ve druhé části této kapitoly se podíváme, jakým způsobem můžeme SVG kód vkládat na webové stránky (studijní článek SVG v XHTML a HTML5) a pak přistoupíme k rozpohybování dosud statických obrázků. Ve studijním článku SVG animace se seznámíme s možnostmi, jak tyto animace tvořit a v dalších studijních článcích se seznámíme se zápisem základních animačních elementů a jejich atributů. Poté následuje ukázka a vysvětlení zápisu SVG kódu animace a příklady doplněné o ukázky animací. Kapitulu ukončuje autotest a pár úkolů na procvičení zápisu SVG animací.

Třetí kapitola *Animace v editoru* seznámí studenty s prostředím editoru a na příkladech si budou moci vyzkoušet vlastní tvorbu animací. Studijní články jsou vhodně doplněny o úkoly, které studenty kurzu průběžně učí práci v editoru Animatron. Kapitola bude mít celkem deset studijních článků a 10 úkolů.

2.3 TVORBA STUDIJNÍCH ČLÁNKŮ

Tato kapitola obsahuje náplně studijních článků, které budou součástí realizace výukového kurzu.

2.3.1 ANIMACE

2.3.1.1 Definice animace

Viz. 1.1 Definice.

Viz. 1.2 Historie a vývoj animace.

2.3.1.2 Princip animace

Viz. 1.3 Princip animace.

2.3.1.3 Fáze tvorby animace

Než se pustíme do vlastní tvorby animací, měli bychom si rozmyslet, co vlastně chceme tvořit a co všechno k tomu budeme potřebovat.

Proto je vhodné sestavit si scénář animace na základě našeho námětu. Například takto:

1. Rozmyslíme si, jak bude naše animace vypadat. Můžeme si napsat jednotlivé scény, které v animaci budou a ty ještě rozdělit na jednotlivé záběry.
2. Rozmyslíme si, jak bude vypadat prostředí animace (pozadí, postavy apod.) a které objekty budou animovány.
3. Objekty potřebné k animaci si vytvoříme (v animačním nebo jiném editoru) a připravíme i další multimediální podklady.
4. Vyskládáme si objekty, které budou tvořit pozadí a objekty, které budou statické. Objekty, které se nebudou animovat, je vhodné uzamknout, aby se s nimi nedalo dále pohybovat.
5. Nyní přejdeme k vlastní animaci a vytvoříme si klíčové snímky jednotlivých objektů.
6. Podle potřeby objekty sloučíme, vrstvíme apod.
7. Následují poslední úpravy, stříh apod.
8. Nakonec zkontrolujeme, zda animace dělá to, co jsme si na začátku určili.
9. Animaci vyexportujeme do požadovaného formátu.

2.3.2 SVG ANIMACE

2.3.2.1 Formát SVG

Formát SVG

Zkratka SVG znamená v překladu Škálovatelná vektorová grafika.

- Scalable = možnost libovolného zvětšování a zmenšování bez ztráty kvality.
- Vector Graphics = matematicky definované geometrické objekty, ze kterých jsou SVG objekty tvořené.

Popis

- Jde o značkovací jazyk a formát souboru popisující dvourozměrnou vektorovou grafiku prostřednictvím značkovacího jazyka XML.
- Zároveň je otevřeným standardem vyvinutým konsorciem W3C roku 1999.
- Umožňuje i propojení s rastrovou grafikou, vložení textu.
- Umožňuje i tvorbu animací a interaktivitu.

Pro vytvoření nebo úpravu souboru nám postačí obyčejný textový editor.

Samozřejmě existuje i velké množství editorů, které s formátem SVG umí pracovat (jde o komerční produkty ale i o open source programy).

SVG není jen nástrojem k zobrazení statických obrázků (loga, ikony, prvky uživatelského rozhraní, ilustrace), často se používá pro interaktivní mapové portály (GIS), jednodušší hry, online grafické editory, interaktivní grafické prvky apod.

Objekt vytvořený v SVG

- Je složen ze základních geometrických tvarů (obdélník, kruh, elipsa, čára, lomená čára, mnohoúhelník), cest, textu (není tvořen cestami), podporuje i možnost ořezu, průhlednosti a práce s rastrovou grafikou.
- Jednotlivé objekty mohou být seskupovány, formátovány pomocí atributů nebo CSS apod.
- Objekty můžeme měnit pomocí knihovny DOM, použít CSS styly nebo JavaScript.

Zdrojový kód lze uložit buď jako soubor s příponou .svg, tím vznikne plnohodnotný obrázek nebo můžeme kód vložit do XHTML/HTML5 kódu.

Podpora v prohlížečích

V současnosti už většina webových prohlížečů formát SVG podporuje bez nutnosti instalace zásuvných modulů. Jen u verzí Internet Explorer 8 a nižší a nižší a mobilních operačních systémů Android 2.3 a nižší.

2.3.2.2 Základní struktura SVG kódu [2][3][4]

Základní struktura SVG dokumentu

```
<?xml version=1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
width="1000px" height="1000px" viewBox="0 0 100% 100%"
```

```
preserveAspectRatio="none">
<title>Název</title>
...
</svg>
```

Vysvětlení kódu

`<?xml version=1.0" encoding="UTF-8"?>` – verze XML dokumentu, nastavení kódování

`<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">` – vhodné uvést pro doplnění přednastavených atributů a pro validaci

`<svg xmlns="http://www.w3.org/2000/svg"` – implicitní jmenný prostor (důležité uvést, jinak se obrázek nevykreslí, ale bude to jen kód XML)

`width="1000px"` – šířka kreslicího plátna (jednotky: px, mm, cm atd.)⁷

`height="1000px"` – výška kreslicího plátna (stejně jednotky jako u šířky)

`viewBox="0 0 100% 100%"` – virtuální souřadnice - umožňuje změnit počátek souřadnic a přizpůsobení velikosti okna – "souřadnice x souřadnice y šířka výška" (např. pokud potřebujeme přizpůsobit obrázek velikosti okna prohlížeče)

`preserveAspectRatio="none"` – předchází deformaci zobrazení při nastavení `viewBoxu`, při použití atributu "none" dochází k deformaci souřadnic takovým způsobem, aby obrázek pokryl plochu okna o zadaných rozměrech (`width`, `height`) a při použití atributu `"xMidYMid"` dochází k přizpůsobení obrázku proporcionálně, tzn. tak, že obrázek se nám zobrazí celý a vycentruje se v obou směrech

2.3.2.3 Společné vlastnosti elementů [2][3][4]

Výchozí počátek souřadnic je z horního levého rohu (pokud není nastaveno jinak).

Objekty jsou vykreslovány v takovém pořadí, jak jsou zapsány ve zdrojovém kódu.

Jednotky u atributů

- Pokud uvedeme hodnotu bez jednotky, výchozí jednotkou je pixel (px).
- Možnost použít libovolné jednotky (např. mm, cm, px atp.).

⁷ Pokud jednotky neuvedeme, budou brány automaticky v pixelech (px)

Vlastnosti elementů

stroke

= barva obrysu

- název barvy (např. ="red")
- RGB (např. ="rgb(140,140,140)")
- hexadecimální (např. ="#8C8C8C")

fill

= barvy výplně

- bez výplně ="none"
- název barvy (např. ="red")
- RGB (např. ="rgb(140,140,140)")
- hexadecimální (např. ="#8C8C8C")

stroke-width

= šířka obrysu

- výchozí hodnota je zadávána v pixelech (např. ="1")

Každá výplň a obrys může mít typ zbarvení:

- plnou barvou
- přechodovou barvou (lineární nebo kruhovou)
- texturou

2.3.2.4 Základní elementy

```
<svg> ... </svg>
```

- mezi tyto značky se zapisují další elementy podle zásad značkovacího jazyka XML
- obsahuje všechny atributy, které jsou potřebné ke korektnímu zobrazení
- umožňuje vnořování elementu `svg` do jiných elementů `svg` (např. když je potřeba nastavit vlastní souřadnicový systém)

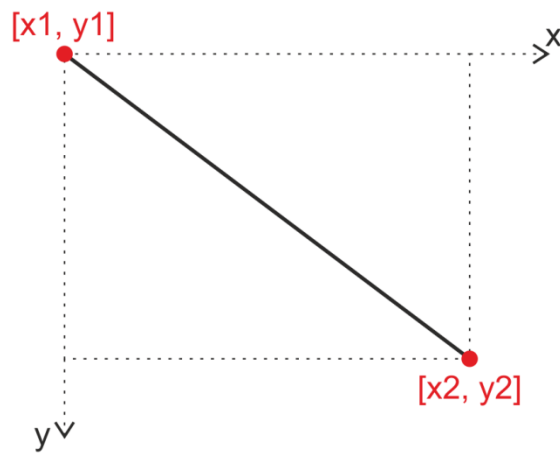
line

= čára, úsečka

- nejjednodušší geometrický tvar
- `x1`, `y1` – souřadnice počátečního bodu

- x_2, y_2 – souřadnice koncového bodu

```
<line stroke = "black" x1="0" y1="0" x2="200" y2="150" />
```



Obrázek 3: Ukázka elementu Line

rectangle

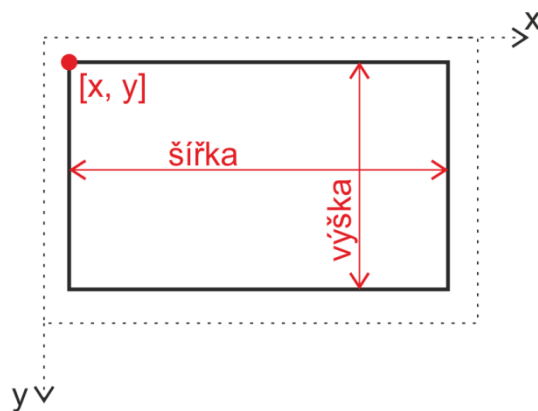
= obdélník

x_1, y_1 – souřadnice počátečního bodu

width – šířka obdélníka

height – výška obdélníka

```
<rect stroke = "black" fill="none" x="10" y="10"
width="50" height="30" />
```



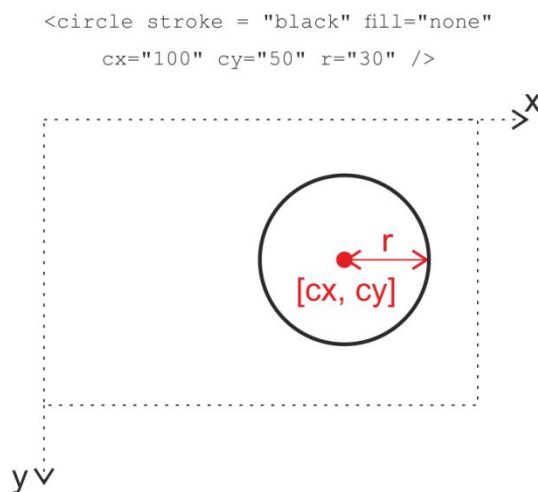
Obrázek 4: Ukázka elementu Rectangle

circle

= kružnice

cx , cy – souřadnice středu kružnice

r – poloměr



Obrázek 5: Ukázka elementu Circle

ellipse

= elipsa

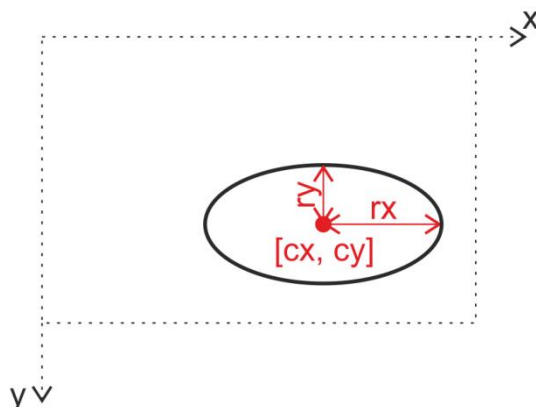
cx , cy – souřadnice středu elipsy

rx – poloměr osy x

ry – poloměr osy y

`transform` – možnost posunu a rotace (`rotate` – úhel rotace (ve stupních), `translate` – posun ve směru x a y)

```
<ellipse stroke = "black" fill="none" cx="150" cy="100"
      rx="100" ry="50" />
```



Obrázek 6: Ukázka elementu Ellipse

polyline

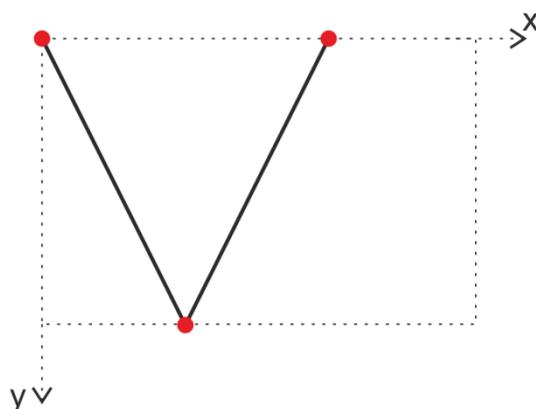
= lomená čára

Je složena z libovolného počtu úseček, které na sebe navazují.

Každá úsečka je zapsána souřadnicemi x, y , které jsou od sebe odděleny čárkou.

Např. Jedna úsečka: `points = "0,0 100,200"` – od bodu x, y do bodu x, y .

```
<polyline stroke = "black" fill="none"
      points="0,0 100,200 200,0" />
```



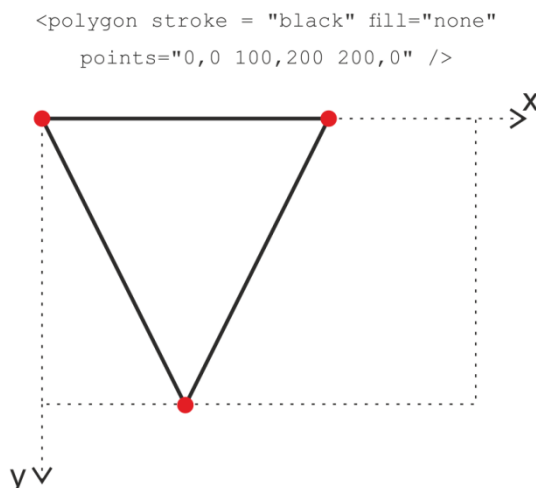
Obrázek 7: Ukázka elementu Polyline

polygon

= mnohoúhelník

Každá úsečka je zapsána souřadnicemi x, y , které jsou od sebe odděleny čárkou.

Koncová souřadnice se vždy spojí s počáteční souřadnicí úsečkou.



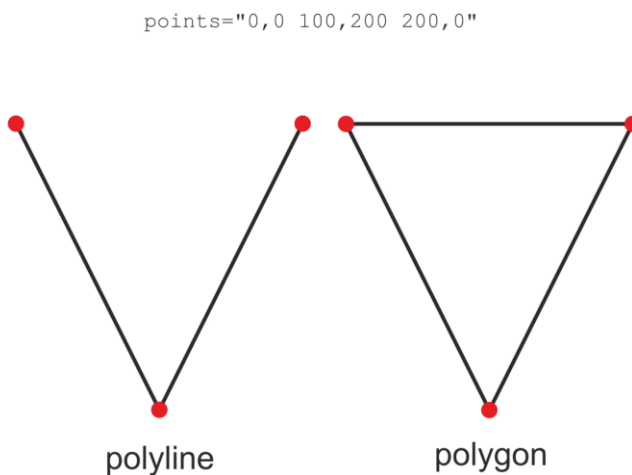
Obrázek 8: Ukázka elementu Polygon

Rozdíl mezi polyline a polygon

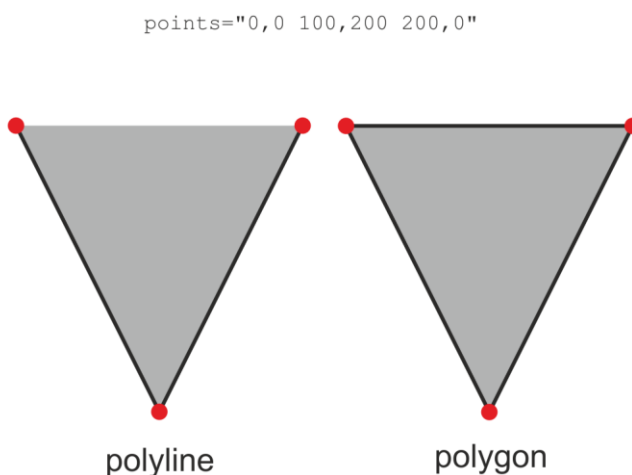
U polyline – pokud nezadáme koncovou souřadnici, která se rovná počáteční souřadnici, objekt bude neuzavřený.

U polygon – koncovou souřadnici uvádět nemusíme, objekt se uzavře automaticky

pokud nastavíme výplň objektu, oba objekty se vyplní, i když polyline nebude uzavřený.



Obrázek 9: Rodíl mezi elementem Polyline a Polygon



Obrázek 10: Rozdíl mezi výplní u elementu Polyline a Polygon

2.3.2.5 Další elementy [2][3][4]

path

= cesta

Element s nejširším využitím, který umožňuje tvořit libovolné tvary kombinací základních elementů a Bézierových křivek.

Příkazy, které slouží k přesouvání počátku kreslení, vytváření úseček a uzavírání cest, se zapisují do atributu `d` `<path d = " " />`

příkazy

- `moveto M/m` – slouží k přesouvání počátku kreslení na danou souřadnici
`M100,200` – přesunutí počátku kreslení na absolutní souřadnici [100, 200]
`m100,200` – přesunutí počátku kreslení o 100 px doprava a o 200 px dolů od aktuální pozice
- `lineto L/l` – slouží k vykreslení úsečky
`L100,-30` – vykreslení úsečky s koncovým bodem na souřadnici [100, -30]
`l100,-30` – vykreslení úsečky s koncovým bodem o 100 px doprava a o 30 px nahoru od aktuální pozice
- `horizontallineto H/h` – slouží k vykreslení vodorovné úsečky

H100 – vykreslení úsečky s koncovým bodem na souřadnici x = 100

h100 – vykreslení úsečky s koncovým bodem o 100 px doprava od aktuální pozice

- `verticalline` `v/v` – slouží k vykreslení svislé úsečky

V100 – vykreslení úsečky s koncovým bodem na souřadnici y = 100

v100 – vykreslení úsečky s koncovým bodem o 100 px dolů od aktuální pozice

- `closepath` `z/z` – slouží k vykreslení úsečky z aktuální pozice souřadnic do počáteční pozice souřadnic
- `Q/q` – slouží k vykreslení kvadratické Bézierovy křivky
- `T/t` – slouží k vykreslení navazující kvadratické Bézierovy křivky
- `C/c` – slouží k vykreslení kubické Bézierovy křivky
- `S/s` – slouží k vykreslení navazující kubické Bézierovy křivky
- `A/a` – slouží k vykreslení eliptické výseče

Příklad 1

Rozdíl mezi `<path fill="none" stroke="black" stroke-width="5" d="M100,100 H200 V200 H100"/>` a `<path fill="none" stroke="black" stroke-width="5" d="M100,100 H200 V200 H100 z"/>`.



Obrázek 11: Ukázka rozdílu zápisu cesty

Příklad 2

Rozdíl mezi `<path fill="rgb(0,152,70)" stroke="black" stroke-width="5" d="M100,100 H200 V200 H100 z m150,100 l100,-50 l-100,100 z"/>` a `<path fill="rgb(0,152,70)" stroke="black" stroke-width="5" d="M100,100 H200 V200 H100 z"/>` `<path fill="rgb(57,49,133)" stroke="black" stroke-width="5" d="m150,100 l100,-50 l-100,100 z"/>` je takový, že v první situaci oba obrazce tvoří jeden objekt, zato v druhé situaci je každý obrazec jedním objektem.



Obrázek 12: Ukázka rozdílu mezi zápoisem cest

text

Zapíše se mezi párový element `<text> ... </text>`

- `x, y` - počáteční souřadnice
- `font-family` – typ písma
- `font-size` – velikost písma
- `font-style` – řez písma
 - a. `italic`
 - b. `normal`
 - c. `oblique`
- `font-weight` – tloušťka písma
 - `lighter` – tenké
 - `normal` – normální
 - `bold` – tučné
- `bolder` – tučnější
- `text-anchor` – význam zadané souřadnice `x`
 - `start` – výchozí nastavení (z levého rohu)
 - `middle` – ze středu textu
 - `end` – z pravého rohu

skupina

- Několik objektů můžeme seskupit do skupiny.
- Dané objekty se zapisují mezi párový element `<g> ... </g>`
- Výhodou toho je, že pokud chceme změnit vlastnosti všech objektů, změníme vlastnosti jen dané skupiny a nemusíme je měnit u všech objektů zvlášť.

2.3.2.6 SVG v XHTML a HTML5 [2][4]

Existují dva způsoby, jak vložit vytvořené objekty SVG do XHTML nebo XHTML5 souboru:

- **SVG kód vložit přímo do zdrojového kódu souboru**

U XHTML realizujeme vložení SVG kódu pomocí dvojtečkové notace - do každého SVG kódu musíme přidat `svg`

Složitější úprava na každé stránce, kde SVG kód použijeme.

Prohlížeče vyžadují přípony `.xml` nebo `.xhtml` pro správné zobrazení (problém může nastat u starších verzí prohlížeče Internet Explorer, který vyžaduje příponu `.htm` nebo `.html`).

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:svg="http://www.w3.org/2000/svg">
  <head>
    <object></object>
  </head>
  <body>
    <svg:svg width="100" height="100">
      <svg:rect width="100" height="100" fill="green"/>
      <svg:text y="50" x="50" text-anchor="middle"
        font-size="35" fill="black">SVG</svg:text>
    </svg:svg>
  </body>
</html>
```

Obrázek 13: Ukázka vložení SVG kódu přímo do zdrojového kódu webových stránek XHTML

U HTML5 realizujeme vložení SVG kódu bez dvojtečkové notace.

Jednodušší vložení bez nutnosti dalších úprav SVG kódu.

Prohlížeče vyžadují přípony `.html`

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
  </head>
  <body>
    <svg width="100" height="100">
      <rect width="100" height="100" fill="green"/>
      <text y="50" x="50" text-anchor="middle"
        font-size="35" fill="black">SVG</text>
    </svg>
  </body>
</html>
```

- **SVG soubor vložit do zdrojového kódu webových stránek podobně jako např. obrázek**

Realizace pomocí tagu `object` (problém může nastat u starších verzí prohlížeče Internet Explorer a proto je v tomto případě vhodné použít tag `embed`)

Univerzální použití.

Méně komplikované.

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:svg="http://www.w3.org/2000/svg">
  <head>
  </head>
  <body>
  <object data="test.svg" type="image/svg+xml" width="100" height="100" />
  </body>
</html>

test.svg
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xml:space="preserve"
  width="100" height="100" version="1.1" shape-rendering="geometricPrecision"
  text-rendering="geometricPrecision" image-rendering="optimizeQuality"
  fill-rule="evenodd" clip-rule="evenodd" xmlns:xlink="http://www.w3.org/1999/xlink">
<rect width="100" height="100" fill="green"/>
<text y="50" x="50" text-anchor="middle" font-size="35" fill="black">SVG</text>
</svg>
```

Obrázek 14: Ukázka vložení SVG souboru do zdrojového kódu webových stránek

2.3.2.7 SVG animace

V předchozích studijních článcích jsme se seznámili s tvorbou statických objektů pomocí SVG kódu.

V následujících článcích se seznámíme s tím, jak tyto statické objekty oživit.

Abychom mohli tvary vytvořené v SVG rozpohybovat, nabízí se nám několik způsobů:

- Animace v SVG jsou **deklarativní**, tzn., že vždy přesně definujeme, který atribut daného elementu bude animován. Tento způsob animace je kompatibilní se standardem **SMIL**. Proto nám stačí určit počáteční a koncový bod daného atributu, barvy, pohybu nebo transformace, způsob, jakým má být animace spuštěna a dobu jejího trvání. Jde o velmi jednoduchý postup, který nevyžaduje znalosti skriptování.
- Další možností, jak vytvářet animace, je použitím knihovny DOM. Jde o přiřazování událostí jednotlivým objektům dokumentu (elementy, atributy, text a komentáře) prostřednictvím Java Scriptu. Tento způsob umožňuje velmi rozsáhlé možnosti, jak vytvořit efektní animace téměř jakéhokoliv druhu.

2.3.2.8 Základní animační elementy

Každý element může animovat jen jeden atribut daného objektu.

Pokud chceme vytvořit animaci více atributů v rámci jednoho elementu, musíme použít více deklarací.

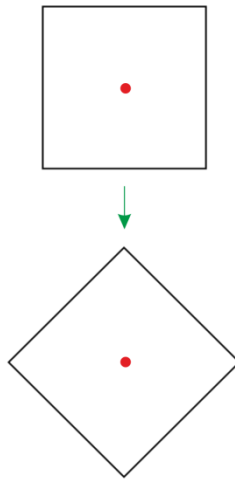
Formát SVG obsahuje některé animační elementy, které jsou definované ve standardu

SMIL:

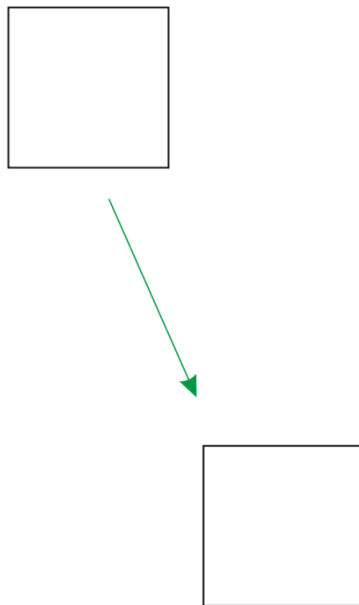
- animate - provádí změnu číselných atributů v závislosti na čase
- set - provádí změnu nečíselných atributů (text apod.) v závislosti na čase
- animateMotion – provádí pohyb daného objektu (resp. celých skupin objektů, vložených obrázků apod.) po určité křivce, která je typu path
atributy elementu (více podrobností k elementům viz 2.3.2.9 *Atributy animací*):
 - i. mpath – určení křivky, po které se bude animovaný objekt pohybovat (umísťuje se do elementu animateMotion a má v tomto případě větší prioritu než element path)
 - ii. d="data_křivky" – specifikace tvaru křivky
 - iii. calcMode
 - iv. keyPoints – určeny společně s atributem keyTimes pro určení klíčových bodů
 - v. keyTimes
 - vi. rotate
 - vii. from, to, by, values
 - viii. xlink:href="#jmeno"
- animateColor – provádí změnu barvy (místo číselných hodnot se uvádí barva podle pravidel společných vlastností elementů viz 2.3.2.3 *Společné vlastnosti elementů*)

Dále jsou to elementy, které jsou specifické pouze pro SVG animace:

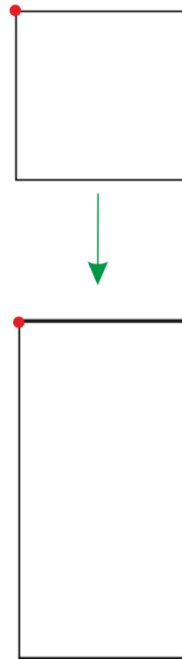
- animateTransform - provádí transformaci daného objektu
Transformace mohou být typu: rotace (rotate), změny polohy (translate), změny velikosti (scale), zkosení (skewX nebo skewY) a jejich vzájemných kombinací.



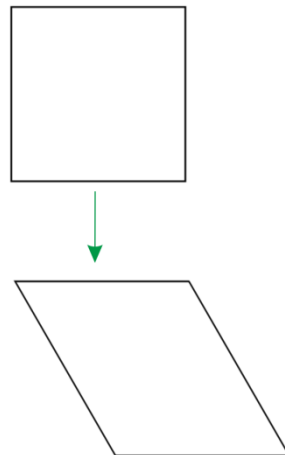
Obrázek 15: Ukázka rotace objektu



Obrázek 16: Ukázka transformace objektu



Obrázek 17: Ukázka změny měřítka objektu



Obrázek 18: Ukázka zkosení objektu

2.3.2.9 Atributy animací [2][3][4]

Pro určení animovaného atributu nám slouží atributy:

- `attributeName` – určení atributu, který chceme následujícím elementem animovat, např. atribut `r` pro změnu poloměru elementu `circle` (kružnice)
- `attributeType` – určení jmenného prostoru, ve kterém je daný atribut definován
CSS – typ atributu odpovídá názvu vlastnosti z CSS stylů
XML – typ atributu odpovídá názvu atributu jazyka XML

auto (výchozí hodnota) – nejprve se prohledá seznam vlastností CSS stylů pro odpovídající název vlastnosti atributu a pokud nebyl nalezen, je zařazen do vlastností atributů jazyka XML

Cílový prvek (element), který chceme animovat, je definován:

- a. explicitně – použitím atributu `<xlink:href="#jmeno" ...>` je odkázáno na cílový element, který bude animován (cílový element pojmenujeme např. `<rect id="obdelnik1" ...>`)
- b. implicitně – cílový element animace se váže na nadřazený prvek

Pro určení rozsahu hodnot atributů nám slouží následující atributy:

- from – zadání počáteční hodnoty animovaného atributu
- to – zadání koncové hodnoty animovaného atributu
- by – celková hodnota změny animovaného atributu (v tomto případě už nemusíme nastavovat hodnotu atributu to)
- values – určení seznamu hodnot, kterých bude daná animace nabývat (v tomto případě se atributy from, to a by budou ignorovat), např. `<animate attributeName="width" values="10;50;10" dur="5s" />` - změna šířky z 10 na 50 a zpět na 10

Pro určení dalších podmínek animace nám slouží atributy:

- begin – určení způsobu, jakým bude animace spuštěna
 - offset-value - určení doby od načtení, kdy bude animace spuštěna (např. `begin="5s"`)
 - syncbase-value - určení, kdy bude animace spuštěna s ohledem na jinou animaci (např. `begin="id_jineanimace.start"` – animace bude spuštěna při startu jiné animace, `begin="id_jineanimace.end"` – animace bude spuštěna po skončení jiné animace, `begin="id_jineanimace.start+10s"` – animace bude spuštěna 10 sekund po startu jiné animace apod.)
 - event-value – určení, jakým typem události bude animace spuštěna (např. `begin="id_jineanimace.mouseover"` – animace bude spuštěna kliknutím myši)
 - repeat-value – určení, kdy se animace spustí s ohledem na počet opakování dané cyklické animace (např. `begin="id_cyklickaanimace.repeat(2)"` – animace bude spuštěna po dvojnásobném opakování dané cyklické animace)

- `accessKey-value` – určení, kterou klávesou bude daná animace spuštěna (např. `begin="id_anim.accessKey('A')"` – animace bude spuštěna po stisku klávesy A)
- `wallclock-sync-value` – určení, kdy se daná animace spustí (řízeno podle reálného času)
- `indefinite` – určení, že se animace spustí voláním funkce `beginElement()` nebo odkazem na animační element
- `end` – určení způsobu, jakým bude animace ukončena (*parametry jsou totožné s parametry u atributu begin*)
- `restart` – určení způsobu, jakým se bude daná animace restartovat
 - `always` – animaci lze restartovat kdykoliv (výchozí hodnota)
 - `whenNotActive` – animace lze restartovat kdykoliv, pokud není aktivní
 - `never` – znemožnění restartu dané animace
- `dur` – určení doby, po kterou se bude daná animace provádět (např. `dur="5s"` – animace se provede za dobu 5 sekund). Hodnota musí být větší než 0!
- `fill` – určení způsobu, jakým se daná animace bude zachovávat
 - `freeze` – výsledek animace zůstane zachován
 - `remove` – výsledek animace bude odstraněn a daná animace se vrátí do svého původního stavu
- `additive` – určení způsobu, jakým bude animace nahrazena
 - `replace` (výchozí hodnota) – klidová hodnota animace (počáteční) bude nahrazena aktuální hodnotou dané animace
 - `sum` – ke klidové hodnotě animace se přičte aktuální hodnota dané animace
- `accumulate` – určení počáteční hodnoty dané animace probíhající v cyklu opakování
 - `sum` – hodnota z konce předcházejícího cyklu animace se přičte k aktuální hodnotě dané animace
- `repeatCount` – určení počtu opakování dané animace
 - číslo – konkrétní hodnota počtu opakování
 - `indefinite` – nekonečný počet opakování
- `repeatDur` – určení doby, po kterou se daná animace bude opakovat
 - číslo – konkrétní doba, po kterou se bude animace opakovat
 - `indefinite` – nekonečná doba opakování
- `type` – určení typu transformace (resp. kombinace transformací) animačního elementu `animateTransform`

- `translate(x,y)` – určení posunu daného objektu o hodnoty x , y
- `scale(x,y)` – určení měřítka (zvětšení/zmenšení) všech vlastností daného objektu
- `rotate(úhel)` – určení úhlu, o který se daný objekt otočí
- `skewX(úhel)`, `skewY(úhel)` – určení úhlu zkosení daného objektu podél osy x a podél osy y
- `calcMode` – určení způsobu, jakým budou hodnoty interpolovány
 - `linear` – určení lineárního přechodu hodnot (výchozí hodnota)
 - `discrete` – určení skokové změny hodnot mezi klíčovými body
 - `paced` – určení konstantní rychlosti celé doby animace
 - `spline` – určení časového průběhu, který je definován Bézierovou kubickou křivkou (zadáním klíčových řídících bodů)
- `keyTimes` – určení rychlosti změny jednotlivých hodnot v atributu `values`
 Udává se v desetinných číslech od hodnoty 0 (počátek animace) do hodnoty 1 (konec animace).

Pokud je atribut `calcMode="paced"`, jsou tyto nastavené hodnoty ignorovány.

- `keySplines` – doplňková funkce k atributu `keyTimes`. Určení sady čtyř řídících bodů (x_1 y_1 x_2 y_2) v rozmezí od hodnoty 0 do 1 (určení změny hodnot mezi dvěma klíčovými body pomocí Bézierových kubických křivek). Nutnost nastavení atributu `calcMode="spline"`.
- `keyPoints` – určení dálky (klíčových bodů), ve které se objekt nachází v daném čase, který je uložen v atributu `keyTimes`
- `path` – určení cesty, která určuje pohyb daného objektu v animaci
- `rotate` – určení způsobu, jakým se daný objekt bude otáčet na dané cestě (`path`)
 - `auto` – otáčení ve směru tečny dané cesty
 - `auto-reverse` – objekt se otočí o 180° a dále se bude otáčet ve směru tečny dané cesty
 - `úhel` – určení úhlu otáčení

2.3.2.10 Zápis kódu animací [2][3][4]

Z dvou předchozích studijních článků budeme používat základní animační elementy a jejich atributy.

Použijeme i elementy pro vytvoření objektů z prvních studijních článků této kapitoly.

Zápis animace:

```
<rect stroke="black" stroke-width="2" fill="none" x="10" y="10"
width="200" height="200"> [1]
<animate [2]
attributeName="height" [3]
from="100" to="50" [4]
begin="0s" dur="10s" [5]
fill="freeze" /> [6]
</rect> [7]
```

Vysvětlení zápisu:

1. Vykreslení čtverce (`rect`) se černým obrysem, bez výplně, šířkou 200 px a výškou 200 px. Element `rect` neukončujeme, protože animace, které se vztahují k tomuto objektu, se zapisují do párového elementu tohoto objektu.
2. Základní animační element, který zajistí, že se bude daný animovaný element měnit v závislosti na času.
3. Pojmenování atributu čtverce, který chceme animovat. V tomto případě výška.
4. Počáteční hodnota atributu (výška 100 px) a koncová hodnota atributu (výška 50 px).
5. Počátek animace (animace začne ihned po načtení) a potrvá deset sekund.
6. Výsledek animace chceme zachovat. Pokud bychom tento atribut animace neuvedli, hodnota animovaného atributu by se vrátila do své počáteční hodnoty.⁸
7. Uzavření elementu `rect`.

Obrázek 19: Počáteční stav elementu `rect`Obrázek 20: Koncový stav elementu `rect`

⁸ Neplést si `fill` (v rámci animací je atributem SMIL a znamená způsob zachování výsledku animace a v případě SVG vlastností znamená výplň elementu).

2.3.2.11 Příklady zápisu animací

Příklad 1: Zápis času

`begin="wallclock(08:00)"` – animace se spustí v 8 hodin ráno na místě, kde je dokument spuštěn

`begin="wallclock(2016-06-30T12:30+01:00)"` – animace se spustí ve 12:30 v časovém pásmu Středoevropského času

Příklad 2: From, to, by a value

```
<rect ...>
<animate attributeName="width" values="40;100;40" dur="10s" />
</rect>
```

```
<rect ...>
<animate attributeName="width" from="50" to="100" dur="10s" />
</rect>
```

```
<rect ...>
<animate attributeName="width" from="50" by="25" dur="10s" />
</rect>
```

```
<rect ...>
<animate attributeName="width" to="100" dur="10s" />
</rect>
```

```
<rect ...>
<animate attributeName="width" by="30" dur="10s" />
</rect>
```

Příklad 3: Fill

```
<rect ...>
<animate attributeName="width" begin="5s" dur="10s" by="100"
fill="freeze" />
</rect>
```

```
<rect ...>
<animate attributeName="width" begin="5s" dur="10s" by="100" />
</rect>
```

Příklad 4: repeatCount a repeatDur

Za 3 sekundy se animace zopakuje 2krát => celková doba animace bude 6 sekund:

```
<rect ...>
<animate attributeName="width" from="0" to="100" dur="3s"
repeatCount="2"/>
</rect>
```

Za 3 sekundy se animace zopakuje 2,5krát => celková doba animace bude 7,5 sekund:

```
<rect ...>
<animate attributeName="width" from="0" to="100" dur="3s"
repeatCount="2.5"/>
</rect>
```

Animace se bude opakovat po dobu 9 sekund => 3krát (ekvivalentní k hodnotě repeatCount="3"):

```
<rect ...>
<animate attributeName="width" from="0" to="100" dur="3s"
repeatDur="9s"/>
</rect>
```

Nastavení modré barvy po zbytek doby animace:

```
<rect ...>
<animate attributeName="fill" from="red" to="blue" repeatCount="2" />
</rect>
```

```
<rect ...>
<animate attributeName="fill" from="red" to="blue" repeatDur="7s" />
</rect>
```

Příklad 5: Accumulate, Additive

Animace objektu po dané cestě, kdy jeden cyklus trvá 5 sekund a jak se animace opakuje (celkem 4krát) navazuje na předchozí hodnotu a začíná další cestu (stejnou), ale na místě, kde skončil předchozí cyklus animace:

```
<path ... dur="5s" accumulate="sum" repeatCount="4" />
```

Animace

```
<rect x="10" y="50" width="100" height="20" stroke="black" fill="none">
  <animate attributeName="width" begin="10s" dur="5s"
    values="0;20;10"
      1. additive="sum" 2. additive="sum" 3. additive="replace" 4.
additive="replace"
      1. accumulate="sum" 3. accumulate="sum"
    repeatCount="10"
    fill="freeze"/>
</rect>
```

Rozbor:

values="0;20;10" – počáteční hodnota animovaného atributu, zvětšení šířky atributu o 20 px a koncová hodnota animace bude o 10 px větší, než počáteční hodnota.

ad 1. additive="sum" – klidovou hodnotu animace nahradí aktuální hodnota dané animace.

`accumulate="sum"` – k aktuální hodnotě animace se přičítá hodnota z konce předcházejícího cyklu animace.

I když je počáteční hodnota v atributu `values` 0 px, bude brána počáteční hodnota nastavené šířky 100 px (`additive="sum"`) a bude následovat tento postup:

Prvních 5 sekund animace: $100 + 20 - 20 + 10 = 110$

Dalších 5 sekund animace: (protože `accumulate="sum"`, bereme hodnotu z konce předchozího cyklu animace) $110 + 20 - 20 + 10 = 120$

Posledních 5 sekund animace: $190 + 20 - 20 + 10 = 200$

Výsledný objekt animace bude mít šířku 200 px.

ad 2. `additive="sum"`

`accumulate` není zadáno – aktuální hodnota animace nahrazuje klidovou hodnotu

Opět bude brána počáteční hodnota nastavené šířky (200 px).

Prvních 5 sekund animace: $100 + 20 - 20 + 10 = 110$

Dalších 5 sekund animace: (protože `accumulate` není zadáno, bereme počáteční hodnotu animace) $100 + 20 - 20 + 10 = 110$

Výsledný objekt animace bude mít šířku 110 px.

ad 3. `additive="replace"` – klidová hodnota animace (počáteční) bude nahrazena aktuální hodnotou dané animace.

`accumulate="sum"`

Protože je počáteční hodnota atributu `values` 0 px a `additive="replace"`, je počáteční hodnota nahrazena aktuální hodnotou animace (0 px).

Prvních 5 sekund animace: $0 + 20 - 20 + 10 = 10$.

Dalších 5 sekund animace: (protože `accumulate="sum"`, bereme hodnotu z konce předchozího cyklu animace) $10 + 20 - 20 + 10 = 20$.

Výsledný objekt animace bude mít šířku 100 px.

ad 4. `additive="replace"`, `accumulate` není zadáno.

Počáteční hodnota nastavené šířky bude 0 px tak, jako tomu je v předchozím případě.

Prvních 5 sekund animace: $0 + 20 - 20 + 10 = 10$.

Dalších 5 sekund animace: (protože `accumulate` není zadáno, bereme počáteční hodnotu animace) $0 + 20 - 20 + 10 = 10$.

Výsledný objekt animace bude mít šířku 10 px.

Celková doba trvání animace bude 50 sekund ve všech čtyřech případech.

2.3.3 ANIMACE V EDITORU [15]

2.3.3.1 Seznámení s editorem Animatron

V následujících studijních článcích se seznámíme se základními informacemi, pracovním prostředím, jednotlivými nástroji a základní tvorbou animací v online editoru Animatron.

K čemu se online animační editor Animatron používá?

Je to vektorový grafický online editor pro tvorbu a úpravu dvourozměrných vektorových animací. Umožňuje vkládání audia, videa, obrázků, je vhodný pro tvorbu reklamních bannerů, infografik atp. Díky jeho intuitivnímu pracovnímu prostředí je vhodný i pro úplné začátečníky, kteří s animacemi teprve začínají.

Spuštění editoru

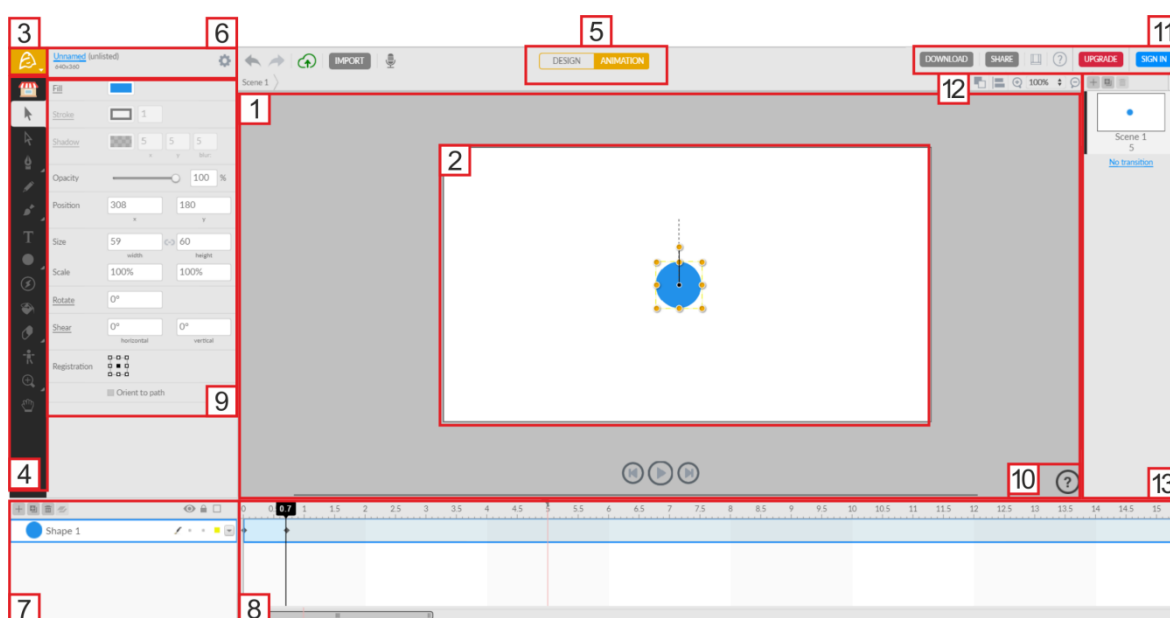
Po zadání webové adresy <http://www.animatron.com> nás přivítá úvodní stránka editoru.

Po kliknutí na **Start Creating** nebo **Start Designing** se nám načte editor, ve kterém budeme tvořit objekty a animovat je. Bez přihlášení jsme v editoru jako hosti, kteří mají omezené možnosti (např. uložit si výslednou animaci budeme moci jen jako statický obrázek ve formátu PNG). Pro povolení dalších možností je nutné se přihlásit. Máme možnost přihlásit se přes náš účet na Googlu, Facebooku, Twitteru, Dribblu nebo si vytvořit vlastní účet přímo na Animatronu zadáním e-mailu, na který nám přijde aktivační e-mail. Tím získáme účet Free, který je bezplatný a nabízí nám možnost vytvořit 5 veřejných projektů, importování obrázků, audia a videa a limitovaný export do HTML5, videa a formátu GIF, SVG, SVG SMIL.

2.3.3.2 Pracovní plocha

Po načtení editoru se nám zobrazí pracovní prostředí v základním uspořádání. Největší částí je samotná *Pracovní plocha*, na které se uprostřed nachází *Plátno*, jehož výchozí rozměr je nastaven na šířku 640 px a výšku 360 px (změnu velikosti můžeme provést ve vlastnostech projektu). Vytvářet a animovat objekty můžeme na celé *Pracovní ploše*, ale po exportu bude výstupem obsahu jen část, která se nachází na *Plátně*.

Popis pracovního prostředí



Obrázek 21: Popis pracovní plochy Animatronu

1. Pracovní plocha

2. Plátno

3. Menu – najdeme zde základní funkce:

- Projekt (Nový, Otevřít existující, Duplikovat, Sdílet, Stáhnout jako HTML5, PNG, SVG, SVG SMIL, GIF, Video, Smazat atd.).
- Upravit (Zpět, Znovu, Vymout, Kopírovat, Vložit atd.).
- Zobrazení (Mřížka, Nastavení mřížky, Přichycení k mřížce, Časová osa, Scény atd.).
- Přehrávání (Spustit/Ukončit, Spustit scénu, atd.).

- Nápověda.
 - Importování souborů.
 - Seznam projektů pro rychlé otevření.
4. **Nástroje** – nachází se na levé straně pracovní plochy. Obsahují všechny základní nástroje pro kreslení a úpravu objektů (více viz 2.3.3.4 *Nástroje*).
 5. **Přepínání mezi návrhem a animací** – vytváření a úprava objektů versus animování těchto objektů.
 6. **Vlastnosti projektu** – nastavení vlastností (jméno projektu, velikost plátna, barva pozadí, popis, klíčová slova, přístup atd.). Více viz 2.3.3.3 *Vlastnosti projektu*.
 7. **Vrstvy** – seznam jednotlivých objektů animace.
 8. **Časová osa** – zobrazení aktuálního času animace, editace klíčových snímků.
 9. **Vlastnosti vybraného objektu** – okno se mění v závislosti na vybraném objektu a najdeme v něm specifické vlastnosti pro tento objekt (např. barva výplně a obrysu, zarovnání, pozice, rozměry, apod.).
 10. **Nápověda** – pokud si nápovědu zapneme, budou se nám zobrazovat informace u aktuálně vybraných nástrojů a objektů.
 11. **Rychlé volby**
 - Stažení animace (HTML5, PNG, SVG, SVG SMIL, Video, GIF).
 - Sdílení projektu (vygenerování kódu, odkaz, GIF, Video).
 - Možnosti uspořádání pracovní plochy.
 - Uživatelská podpora.
 - Upgrade uživatelského účtu.
 - Přihlášení/Profil uživatele.
 12. Kombinace objektů, Zarovnání objektů, Lupa a procentní velikost zobrazení.
 13. **Scény** (přidat, odebrat, duplikovat atd.).

2.3.3.3 Vlastnosti projektu

Obrázek 22: Vlastnosti projektu

1. Zobrazení okna vlastností aktuálního projektu.
2. **Jméno** (*Name*) – pojmenování projektu.
3. **Rozměry plátna** (*Size*) – výchozí hodnotou je šířka 640 px a výška 360 px. Další výběr je možný z rychlé nabídky předvolených rozměrů (*Choose from presets*) nebo zadáním vlastních rozměrů.
4. **Pozadí** (*Background*) – barvu pozadí můžeme namíchat, zadat v hexadecimálním tvaru, v RGB nebo kapátkem. Je zde možnost nastavení i alfa kanálu.
5. **Popis** (*Description*) – místo pro popsání aktuálního projektu, jeho vlastností, cílů apod.
6. **Klíčová slova** (*Tags*) – zapsání klíčových slov charakterizujících daný projekt. Jednotlivá slova oddělujeme čárkou a povolena jsou písmena, číslice, pomlčky a podtržítka.

7. **Přístup k projektu** (*Project Access*) – nastavení typu projektu (veřejný/neurčeno/privátní). V případě uživatelské účtu FREE je možné nastavit jen veřejný projekt, který je viditelný a dohledatelný na serveru Animatronu. Dále můžeme ještě povolit, zda umožníme projekt upravovat ostatním uživatelům nebo ne (*Modify Project*). Další možnosti přístupu k projektu jsou dostupné pouze v placených uživatelských účtech PRO a BUSINESS.

2.3.3.4 Nástroje

V tomto studijním článku se seznámíme s nástroji, jejich funkcemi a vlastnostmi. Některé z nich jsou totožné s nástroji ve vektorových editorech. Pomocí nich budeme vytvářet vektorové objekty, které budeme animovat.

Výběr (*Selection*)



Obrázek 23: Ikona nástroje Výběr

Umožňuje výběr objektu nebo skupiny více objektů. Tím, že objekt označíme, zobrazí se nám okno s vlastnostmi objektu.



Obrázek 24: Vlastnosti vybraného objektu

1. Barva výplně (*Fill*)
2. Barva a tloušťka obrysu (*Stroke*)
3. Nastavení stínu (*Shadow*)
4. Průhlednost (*Opacity*)
5. Souřadnice pozice (*Position*)
6. Rozměry (*Size*)
7. Nastavení měřítka (*Scale*)
8. Nastavení rotace (*Rotate*)
9. Nastavení zkosení (*Shear*)
10. Nastavení středu objektu (*Registration*)
11. Při umístění objektu na cestu natáčet podle tvaru cesty (*Orient to path*)

Přímý výběr (*Direct Selection*)



Obrázek 25: Ikona nástroje Přímý výběr

Umožňuje úpravu cest a křivek pomocí editace uzlů. Uzel přidáme dvojitým poklepáním na daném místě a odebereme ho dvojitým poklepáním na konkrétní uzel.

Pero (*Pen*)



Obrázek 26: Ikona nástroje Pero

Tento nástroj nám umožňuje nakreslit rovné čáry, ale i křivky.

Pro nakreslení rovné čáry stačí klikat myší na různá místa na plátně a pro nakreslení křivky musíme kliknout myší na dané místo, držet tlačítko myši a táhnout. Délkou modré úchytky kontrolujeme poloměr vytvářeného oblouky křivky a směrem tažení udáváme směr. Křivku ukončíme klepnutím na jiný nástroj.



Obrázek 27: Rozšířená nabídka nástroje Pero

Rozšířená nabídka nástroje Pero nám nabízí:

- Přidání uzlu (*Add Node*)
- Odebrání uzlu (*Remove Node*)
- Převedení uzlu na ostrý úhel (*Convert Node*)

Tužka (*Pencil*)



Obrázek 28: Ikona nástroje Tužka

Tento nástroj je nejjednodušším způsobem, jakým lze vytvořit cestu. Stačí držet tlačítko myši a tažením nakreslit požadovaný tvar cesty. Cestu ukončíme uvolněním tlačítka myši.



Obrázek 29: Vlastnosti nástroje Tužka

1. Barva a tloušťka tužky (*Stroke*)
2. Průhlednost (*Opacity*)
3. Vyhlazení (*Smoothness*) – umožňuje nastavení plynulosti výsledné cesty (čím větší hodnota, tím více bude cesta vyhlazená)

Štětec (*Brush*)



Obrázek 30: Ikona nástroje Štětec

Tento nástroj umožňuje vytvořit uzavřenou křivku. Pokud budeme chtít upravit tvar cesty výsledné křivky, dá nám to mnohem více práce, protože výsledná křivka je složena z jednotlivých segmentů⁹ oddělených jednotlivými uzly.

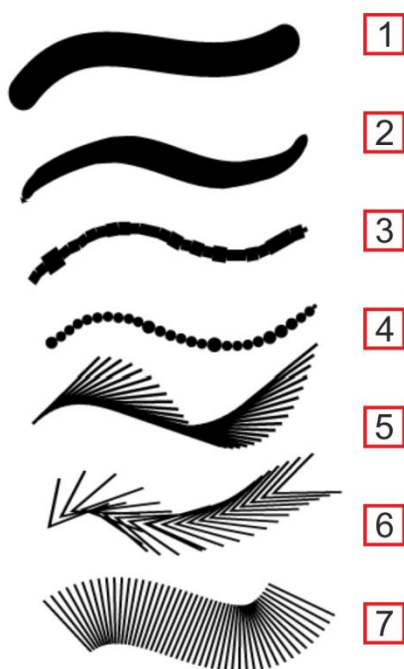


Obrázek 31: Uzavřená křivka



Obrázek 32: Rozšířená nabídka nástroje Štětec

Rozšířená nabídka štětců nám nabízí sedm přednastavených typů štětců:



Obrázek 33: Ukázka všech typů štětců

1. Základní štětec (Brush)
2. Umělecký štětec (Artistic Brush)
3. Čtvercový štětec (Squares Brush)
4. Kruhový štětec (Circles Brush)
5. Štětec se vzhledem peří (Feather Brush)

⁹ Segment = část křivky mezi dvěma uzly.

6. Štětec se vzhledem jehličí (Conifer Brush)

7. Štětec se vzhledem kolejnic (Rails Brush)

Text



Obrázek 34: Ikona nástroje Text

Tento nástroj umožňuje vytvořit a upravovat text.

Základní tvary (*Oval, Rectangle, Star, Polygon, Line*)



Obrázek 35: Ikona nástroje Základní tvary

Tento nástroj umožňuje vytvořit základní tvary. Je nejrychlejším a nejuniverzálnějším nástrojem, jak vytvořit požadovaný objekt.



Obrázek 36: Rozšířená nabídka nástroje Základní tvary

V rozšířené nabídce najdeme:

- Elipsa (Oval) – lze nastavit obrys, výplň a procento průhlednosti
- Čtverec (Rectangle) – lze nastavit obrys, výplň, procento průhlednosti a hodnotu zaoblení (od hodnoty -100 do hodnoty 100)



Obrázek 37: Ukázka nastavení zaoblení nástroje Čtverec

- Hvězda (Star) – lze nastavit obrys, výplň, procento průhlednosti, počet vrcholů a vnitřní poloměr
- Mnohoúhelník (Polygon) – lze nastavit obrys, výplň, procento průhlednosti a počet vrcholů
- Čára (Line) – lze nastavit obrys, výplň a procento průhlednosti

Akce a události (*Actions*)



Obrázek 38: Ikona nástroje Akce a události

Tento nástroj umožňuje jednotlivým objektům přiřazovat určité akce a události (interaktivní funkce), které nám, např. při kliknutí myši na daný objekt, umožní otevření zadaného odkazu.

- Událostí (*Events*) určíme, jaký typ interakce bude použit u daného objektu (např. kliknutí na objekt, kliknutí myši, přejetí myši apod.).
- Akce (*Action*) určí, co se stane, když nastane daná událost (např. po kliknutí na daný objekt se otevře zadaný odkaz apod.).

Cílem tohoto nástroje je zavedení interaktivních funkcí, které posunou výslednou animace o další krok vpřed.

Výplň (Bucket)



Obrázek 39: Ikona nástroje Výplň

Umožňuje vyplnit nastavenou barvou výplně uzavřené oblasti. Každá takto vyplněná oblast tvoří nový, samostatný objekt.

Gumování (Eraser)



Obrázek 40: Ikona nástroje Guma

Držením tlačítka myši a tažením smažeme všechno, přes co myši přejedeme. Můžeme si nastavit i poloměr gumování.

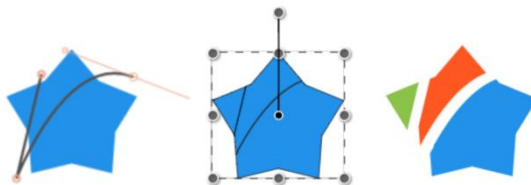


Obrázek 41: Rozšířená nabídka nástroje Guma

Kromě gumování nám rozšířená nabídka nabízí ještě další nástroje:

- Laso (*Lasso*) – kliknutím a nakreslením oblasti od ruky vybereme oblast, kterou budeme chtít smazat. Uvolněním myši se kreslená oblast automaticky uzavře. Klávesou *Delete* oblast smažeme.

- Nůž (*Knife*) – klikáním na různá místa umístíme body a mezi nimi bude rovná čára pro vybíranou oblast. Nebo držením a táhnutím myši vytvoříme zakřivenou čáru pro výběr. (Funkce kreslení oblasti je totožná s kreslením nástrojem *Pero*). Po nakreslení požadované řezné křivky klikneme vlevo na tlačítko *Slit* a tím se nám rozdělí daný objekt (objekty) na více objektů podle řezné křivky.



Obrázek 42: Ukázka nástroje Nůž

Kostra (Bone)



Obrázek 43: Ikona nástroje Kostra

Tento nástroj je velmi užitečný pro oživení objektů postav.

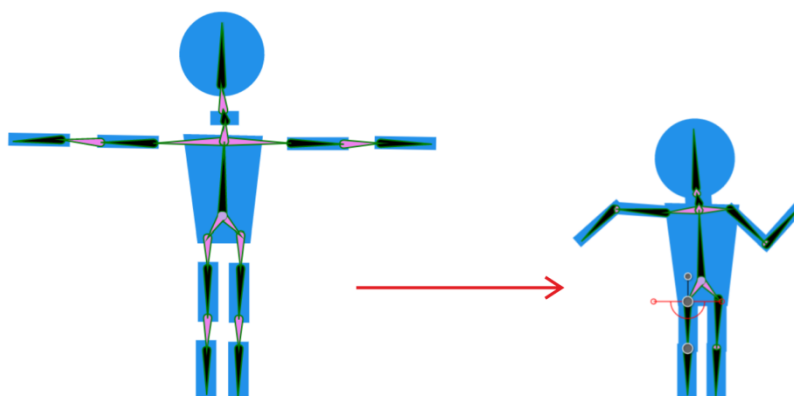
Začneme poskládáním jednotlivých objektů postavy (ruce, nohy, tělo, hlava atd.) tak, aby mezi nimi byla dostatečná mezera. Po vybrání nástroje *Kostra* můžeme začít kostru zakreslovat na objekty. Začneme na těle, kde bude řídicí centrum postavy. Pokračujeme na konec těla na místo, kde by měla začínat noha. Poté klikneme na začátek horní části nohy a poté zase na konec horní části nohy. Dále pokračujeme opět tak, že klikneme na začátek dolní části nohy a opět na konec. Pro ukončení vykreslování kostry stiskneme klávesu *ESC*. V mezerách, které jsme si nechali mezi jednotlivými částmi těla, je kostra vykreslena růžově. Pokud na tuto růžovou část klikneme a označíme ji, můžeme posunout vytvořený kloub do horního vytvořeného kloubu. Tím spojíme jednotlivé části těla a zařídíme, aby se v tomto místě část mohla ohýbat.

Pokud bychom vykreslili část kostry špatně, můžeme se vrátit o krok zpět klávesami *CTRL + Z*.

Všechny části těla (jednotlivé objekty) se nyní uloží jako jedna celistvá vrstva.

Nastavit aktuální zobrazení postavy umožníme tím, že nástrojem *Kostra* označíme jednotlivé části kostry a u daného kloubu se nám zobrazí dvě červená táhla (kterými definujeme, jak moc se může daný kloub ohýbat) a černé táhlo s šedým kolečkem, kterým

definujeme směr. V režimu *Animace* můžeme s částmi postavy pohybovat nástrojem *Výběr*.



Obrázek 44: Ukázka práce s nástrojem Kostra

Lupa (Zoom)



Obrázek 45: Ikona nástroje Lupa

Tento nástroj nám umožňuje přiblížit a oddálit plátno. Stiskem tlačítka myši a výběru oblasti se nám plátno přiblíží do této oblasti.

Nástroj ruka (Hand Tool)



Obrázek 46: Ikona nástroje Ruka

Tento nástroj je užitečný, pokud máme zvětšené plátno více, než na celou pracovní plochu a potřebujeme se po něm pohybovat.

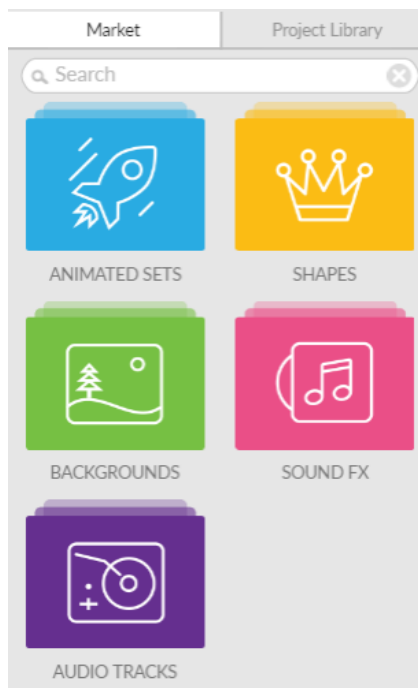
Obchod a knihovna (Market and Project Library)



Obrázek 47: Ikona Obchodu a knihovny

V nabídce nástrojů jako první nalezneme Obchod a knihovnu projektů. Do knihovny projektů si můžeme ukládat vytvořené objekty, animované klipy apod. a použít je např. v dalších projektech.

V obchodě nalezneme spoustu užitečných sad, ze kterých si můžeme tvořit animace.



Obrázek 48: Ukázka nabídky Obchodu

Vybírat můžeme z pěti sad:

- Animované sety (*Animated Sets*) – ty jsou dále rozděleny do kategorií Postavy, Město, Věda, Zvířata atd. Obsahují ukázkou výsledné animace a všechny objekty a klipy, které animace obsahuje. Můžeme použít tyto objekty a klipy k sestavení vlastního projektu.
- Tvary (*Shapes*) – dále jsou také rozděleny do kategorií Objekty, Hry, Příroda atd. Obsahují jen statické objekty, které můžeme přetažením umístit na plátno.
- Pozadí (*Background*) – tato sada obsahuje různá pozadí, která můžeme použít pro naši animaci.
- Zvuky (*Sound FX*) – tato sada obsahuje zvuky, které můžeme přiřadit k objektům.
- Hudba (*Audio Tracks*) – pro naši animaci můžeme použít i hudební podkres.

2.3.3.5 Práce s objekty

Seskupení více objektů

Vytvoříme si například kmen stromu, jeho korunu a pár jablek. Pokud bychom teď chtěli posunout celý strom s jablky doprava a táhli bychom za kmen, posunul by se jen on a zbytek stromu by zůstal na původním místě. Můžeme tedy nástrojem *Výběr* vybrat všechny objekty a posunout je. Jde to ale i jednodušší cestou, pokud tam například

budeme mít více stromů a bylo by obtížnější vybírat tímto způsobem vždy jeden celý strom.

Označíme si tedy všechny objekty (nástrojem *Výběr*), které chceme, aby nám spolu vytvořili jeden objekt.

Po vybrání klikneme pravým tlačítkem myši a z nabídky vybereme možnost *Convert To Movie Clip* (nebo přes nabídku *Menu -> Edit -> Convert To Movie Clip*).

Když nyní zkusíme strom přesunout na jiné místo, vidíme, že strom neztratil ani jedno jablko.

Pokud bychom chtěli strom ještě poupravit, např. odebrat jablka, opět klikneme pravým tlačítkem myši a v nabídce vybereme příkaz *Edit Clip*. Otevře se nám nová záložka jen s objekty, které jsme přidali do skupiny a nyní můžeme upravit, co potřebujeme. Když se pak vrátíme zpět na záložku našeho projektu, uvidíme už všechny změny, které jsme provedli, a strom bude pořád seskupený.

Kombinace objektů

Při práci s objekty se nám můžou hodit operace, které najdeme v pravém horním rohu pod názvem *Kombination*.



Obrázek 49: Kombinace objektů

Najdeme tam operace:

- Sjednocení (*Union*) – umožní nám sečíst vybrané objekty do sebe (není to sloučení objektů, jako v předchozím případě stromu a jablek)
- Průnik (*Intersect*) – umožní nám z vybraných objektů zachování jen té části, ve které se oba objekty překrývali
- Rozdělení na části (*Split*) – umožní nám rozdělení překrývajících se objektů na více objektů podle překrytých částí
- Rozdíl (*Exclude*) – umožní nám odečtení dvou objektů od sebe (horní objekt od dolního)

2.3.3.6 Animování

V předchozích studijních článcích jsme pracovali na vytváření objektů a nyní přejdeme k samotnému animování.

U počítačové animace nám postačí, když nakreslíme první a poslední snímek a dáme počítači pokyn, co má s objektem udělat. On si pak sám dokreslí jednotlivé mezisnímky.

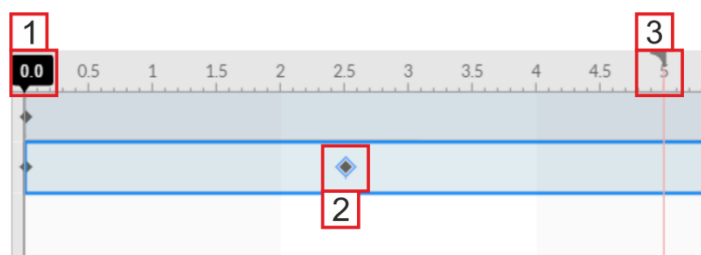
Snímky, které ve skutečnosti nakreslíme, nazýváme klíčové snímky, protože klíčové znamená významné. Pokyny k tomu, co se má mezi klíčovými snímky vykreslit, se nazývají *tweens* (doplnění mezisnímků).

Animatron podporuje čtyři typy vykreslení mezisnímků:

- Posun (*Translate*) – umožňuje přesunutí objektu z jednoho místa na druhé
- Otočení (*Rotate*) – umožňuje otáčení objektu kolem daného bodu
- Měřítko (*Scale*) – umožňuje změnu velikosti objektu
- Průhlednost (*Alpha*) – umožňuje změnu průhlednosti objektu

Abychom se přesunuli do režimu animování, musíme přepnout z DESIGN na ANIMATION.

Časová osa



Obrázek 50: Popis časové osy

1. Přehrávací hlava (*Playhead*)
2. Značka klíčového snímku (*Keyframe Marker*) – šedý kosočtverec
3. Značka konce animace (*End Marker*) – všechny animace se vždy zastaví tam, kde je umístěna tato značka

Přehrávací hlava je nastavena na 0,0 s. To znamená, že aktuálně zobrazený objekt na plátně zobrazuje, jak bude tento objekt vypadat na začátku animace.

Značka klíčového snímku pod přehrávací hlavou značí, že aktuální pozice objektu je klíčovým snímkem.

Klíčové snímky

Když přehrávací hlavu přesuneme po časové ose na jiný čas a přesuneme objekt například na druhou stranu plátna, tak tím dáme pokyn, kde bude daný objekt v jiném čase. Animatron si v tomto čase automaticky vytvoří další klíčový snímek a za objektem se vykreslí přerušovaná šedá čára, která značí cestu, po které objekt „pojede“.



Obrázek 51: Ukázka přesunutí objektu

Když se vrátíme na začátek časové osy a spustíme animaci, objekt se mezi klíčovými snímky plynule přesouvá. A v tom je celá podstata – že Animatron si sám vygeneroval jednotlivé mezisnímky mezi dvěma klíčovými snímky.

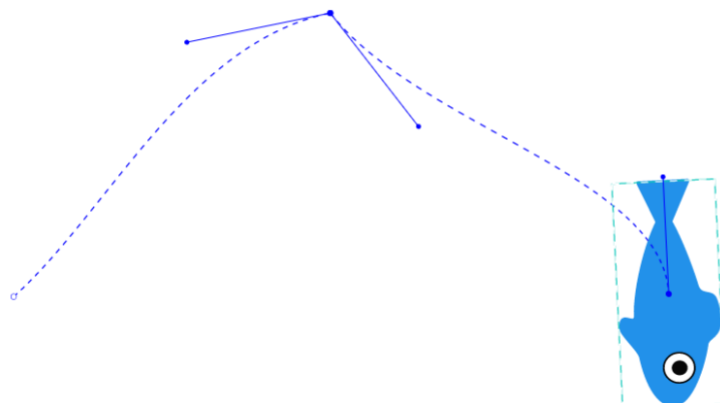
Nyní se můžeme přesouvat na libovolná místa na časové ose mezi těmito klíčovými snímky a uvidíme, co v daný moment objekt dělá.

Přidání klíčových snímků

Pokud budeme chtít, aby objekt při pohybu zleva doprava udělal navíc skok nahoru, budeme postupovat následovně:

- Na časové ose se přesuneme na čas, který se nachází přibližně uprostřed mezi dvěma již existujícími klíčovými snímky.
- Uskutečníme skok tím, že objekt přesuneme nahoru.
- Animatron automaticky vytvořil nový klíčový snímek a cesta pohybu objektu nyní ukazuje, jakým směrem se objekt bude pohybovat.
- Aby se pohyb objektu přizpůsobil tvaru cesty i natáčením, ve *Vlastnostech daného objektu* zaškrtneme možnost *Orient to path*.

Tvar cesty můžeme také upravit pomocí nástroje *Přímý výběr*, stejně tak jako upravujeme křivky, když vytváříme objekty.



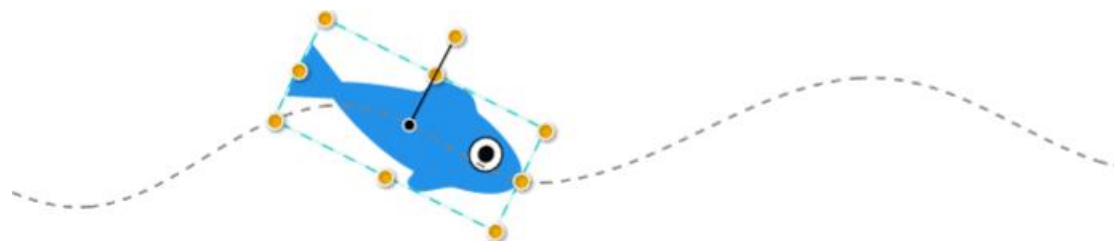
Obrázek 52: Ukázka úpravy cesty animovaného objektu

Existující klíčové snímky můžeme v časové ose také posouvat, pokud jsme se například rozhodli, aby objekt provedl skok později nebo celý průběh pohybu objektu trval delší dobu.

Umístění objektu na cestu

Další způsob, jak objektu definovat směr pohybu, je možnost umístění objektu na cestu definovanou jiným objektem.

- Vybereme, např. nástroj *Tužka* a nakreslíme zleva doprava vlnky, na které chceme umístit rybičku.
- Vytvořenou cestu vyjmeme (pravé tlačítko myši -> *Cut* nebo *CTRL+X*).
- Klikneme pravým tlačítkem na rybičku a z nabídky příkazů vybereme možnost *Snap To Path*. Rybička se nám umístí na cestu a na časové ose se vytvoří klíčové snímky.



Obrázek 53: Ukázka umístění animovaného objektu na cestu

2.3.3.7 Způsoby vykreslování mezisnímků

Posun (Translate)

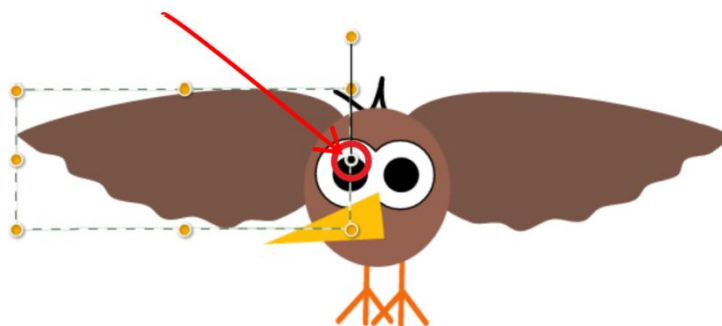
S posouváním jsme se seznámili již v předchozím článku o vytváření klíčových snímků.

Daný objekt stačí v daném čase posunout na určené místo, vytvoří se klíčový snímek a za objektem se vykreslí přerušovaná šedá čára, která definuje cestu, po které se objekt posouvá.

Otočení (Rotate)

Animaci rotace si ukážeme na vytvořeném obrázku ptáčka, který potřebuje rozpohybovat křídla. Aby se křídla otáčeli správně (ne kolem středu celého křídla, jak je nastaveno), musíme nejprve změnit řídicí bod, okolo kterého se bude křídlo otáčet.

- Vybereme jedno křídlo a ve vlastnostech objektu pod *Registration* nastavíme řídicí bod tak, aby se nacházel u těla ptáčka. Rukojeť otáčení si ihned posune na nový řídicí bod. Postup zopakujeme i u druhého křídla.



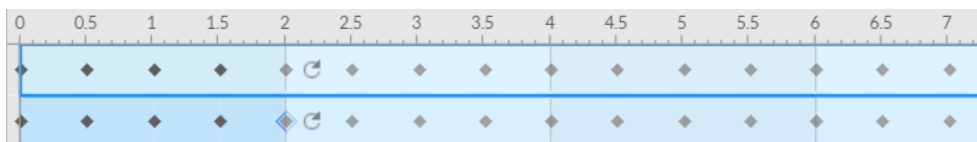
Obrázek 54: Ukázka nastavení řídicího bodu rotace

- V režimu *Animace* si na časové ose najedeme např. na čas 0,5 s a každé křídlo otočíme o několik stupňů. V čase 1,0 s vrátíme křídla do původní polohy, v čase 1,5 s zase každé křídlo otočíme a v 2,0 s ho opět vrátíme do původní polohy. A takto můžeme pokračovat dál.

Opakování části animace

Abychom nemuseli vytvářet desítky až stovky klíčových snímků té samé animace, existuje možnost zadat opakování.

- Smyčka (*Loop*) - pravým tlačítkem myši klikneme na poslední klíčový snímek té části, kterou budeme chtít opakovat (v případě ptáčka tedy např. 2,0 s) a z nabídky vybereme možnost *Change End Action*. Klikneme na *Loop* a zadáme hodnotu 0. Tím dáme pokyn k tomu, že soubor akcí se bude opakovat donekonečna. A ihned vidíme, že Animatron si automaticky vložil klíčové snímky na časovou osu. Postup opět zopakujeme i pro druhé křídlo.



Obrázek 55: Ukázka časové osy při opakování části animace

- Zpětný chod (*Reverse*) – opět pravým tlačítkem myši klikneme na poslední klíčový snímek té části, od které se bude animovat odzadu dopředu. Z nabídky vybereme možnost *Change End Action* a klikneme na *Reverse*. Pokud zadáme hodnotu 0, bude se tato část animovat nepřetržitě, pokud zadáme konkrétní hodnotu, bude se animovat jen tolikrát, kolikrát zadáme.

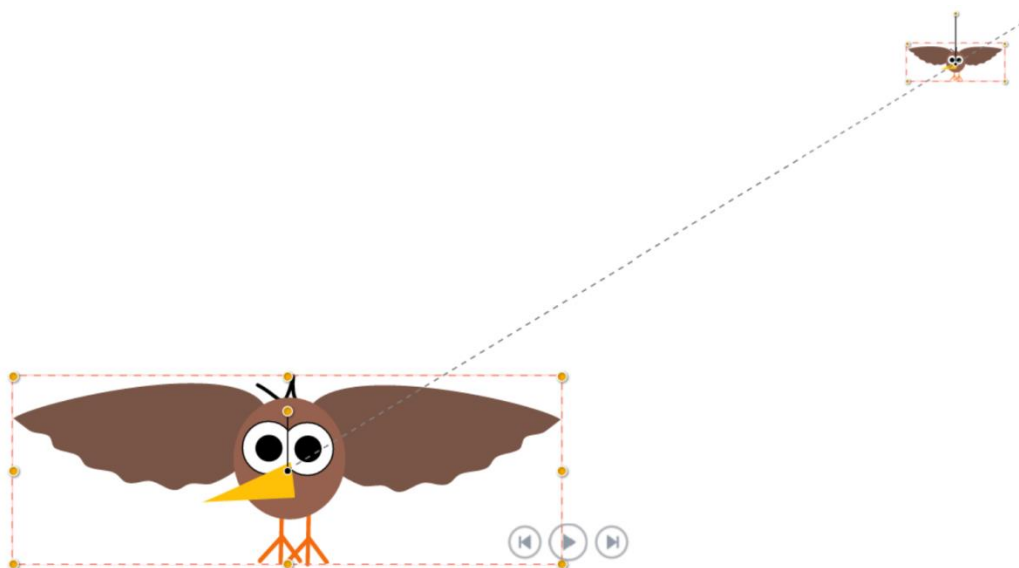
Seskupení objektů včetně animace

Ještě před tím, než se pustíme do dalšího typu vykreslování mezisnímků, seskupíme všechny objekty, které tvoří ptáčka, do jedné skupiny. Nástrojem *Výběr* vybereme všechny objekty a pravým tlačítkem vybereme možnost *Convert To Movie Clip*. Vznikne nám jeden jediný objekt, který má v sobě uloženou i animaci křídel.

Měřítko (Scale)

V dalším případě budeme chtít, aby ptáček odletěl do dálky a postupně se vzdaloval (zmenšoval).

- Na časové ose si najedeme např. na čas 10 s a ptáčka posuneme do pravého horního rohu. Tím nám odletí, ale velikost nezmění.
- Ve vlastnostech pod *Scale* nastavíme u šířky i u výšky hodnotu 10 %.



Obrázek 56: Ukázka nastavení změny měřítka

Průhlednost (*Alpha*)

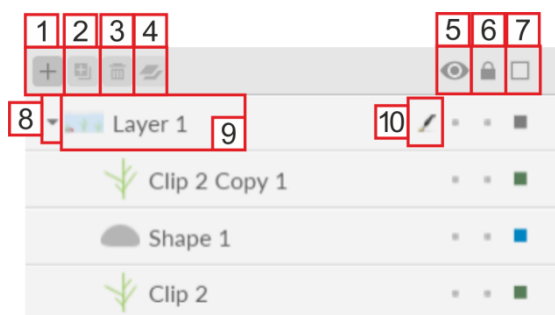
Jak náš animovaný ptáček odlétá do dálky a pomalu mizí z dohledu, nastavíme mu ještě i ubývání viditelnosti. Čím dál bude, tím méně bude viditelný.

Na časové ose se opět přesuneme na čas 10 s a ve vlastnostech pod *Opacity* nastavíme hodnotu 0 %.

2.3.3.8 Vrstvy

Vrstvy tvoří jednotlivé plochy, na kterých jsou jednotlivé objekty nebo jejich části. Vrstvu si můžeme představit jako průhlednou fólii, na kterou něco nakreslíme a pak několik takových fólií dáme na sebe a vznikne nám výsledný obraz.

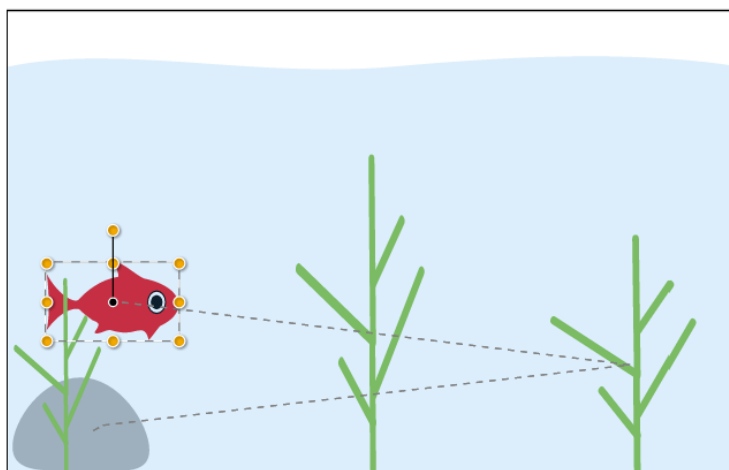
- Pokud chceme vybrat více vrstev najednou, podržíme klávesu *CTRL* a klikáme na jednotlivé vrstvy.
- Vrstvy mezi sebou můžeme prohazovat (přetažením nahoru/dolů) a měnit tak jejich úroveň.
- Vrstvy si můžeme přejmenovat.



Obrázek 57: Popis možností práce s vrstvami

1. Nová vrstva (*Add Layer*) – v podstatě jde jen o složku, ve které budou další vrstvy.
2. Duplikovat vrstvu (*Duplicate Layer*) – vytvoří přesnou kopii vybrané vrstvy nebo více vrstev.
3. Smazat vrstvu (*Delete*) – smaže vybrané vrstvy.
4. Kombinace vrstev (*Combine Layer*) – všechny vybrané vrstvy přidá do jedné složky (v případě více složek sloučí vrstvy do jedné složky).
5. Viditelnost (*Hide all layers*) – skrýt všechny vrstvy (v případě jednotlivých vrstev nastaví viditelnost jen u konkrétní vrstvy).

6. Zamknutí (*Lock all layers*) – zamknout všechny vrstvy (v případě jednotlivých vrstev zamkne jen konkrétní vrstvy). Zamknutím zakážeme jakékoliv operace s daným objektem.
7. Zobrazení obrysu (*Show objects on all layers as outlines*) – zobrazí jen obrysy u všech vrstev (v případě jednotlivých vrstev zobrazí obrysy jen u konkrétní vrstvy).
8. Rozevření složky s vrstvami – seznam nám ukáže, jak jsou objekty vrstvené (např. když kámen je o úroveň výš než rybička, může se rybička za kámen schovat).
9. Složka vrstev – je to užitečný způsob, jakým umístit soubor souvisejících objektů do jednoho místa (např. všechny objekty, které budou v pozadí, přidáme do jedné složky a objekty, které budou v popředí, přidáme do další složky). Složky lze i pojmenovat.
10. Právě vybraná vrstva, na které pracujeme.



Obrázek 58: Ukázka práce s vrstvami

Jak vidíme na příkladu akvářka, do složky *Akvárko* jsme si přesunuli všechny objekty v pořadí: *Nádoba*, *Rostlina*, *Rybička*, *Rostlina*, *Kámen*, *Rostlina*, *Voda*. Všechny objekty kromě rybičky jsme uzamknuli, abychom s nimi už dál nemohli pohybovat.

Budeme pracovat jen s vrstvou *Rybička*. Samotná vrstva *Rybička* je skupinou několika objektů (několik vrstev).

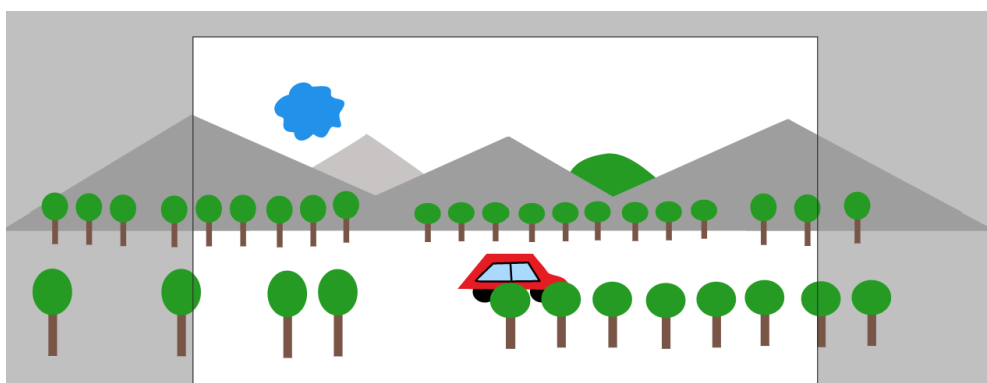
Rybičku umístíme za kámen. Pokud budeme chtít, aby plavala, budeme postupovat jako v předchozích studijních člancích.

Dále si ještě vytvoříme novou složku vrstev (*Add Layers*) a do ní nakreslíme bubliny. Výhodou toho, že máme objekty ve složce je to, že můžeme animovat rovnou celou složku (všechny vrstvy najednou).

2.3.3.9 Další možnosti animování

Panning

Jde o techniku iluze letu kamery, jakoby přejížděla přes scénu, např. sledování auta jedoucího po silnici. Je běžné, že tento efekt se simuluje tak, že předmět zájmu (v našem případě auto) je na stejném místě a pohybují se jen kulisy za ním (stromy, mraky atd.).



Obrázek 59: Ukázka techniky Panning

V režimu návrhu (*Design*) si vytvoříme kulisy:

- Pozadí (hory a mrak) si seskupíme. Stejně tak nakreslíme stromy, které budou v pozadí, a také je seskupíme.
- Nakreslíme si autíčko, jehož objekty také seskupíme, aby tvořilo jeden objekt. Můžeme si nakreslit i objekty, které budou před autem, jen by měly být větší, než v pozadí, aby byla scéna reálnější.

Pozadí a popředí musí přesahovat okraje plátna, abychom je mohli posouvat.

V režimu animace (*Animation*) nastavíme animace jednotlivým objektům:

- Najedeme na počátek animace a všechny skupiny objektů, kromě auta, přesuneme tak, aby jejich levý okraj byl u levého okraje plátna.
- V čase 5 s posuneme stromy v popředí tak, aby pravý okraj byl u pravého okraje plátna.
- To samé uděláme se stromy v pozadí v čase 10 s.
- A v čase 15 s posuneme hory v pozadí o kousek doleva.
- Konec přehrávání animace nastavíme zarážkou na čas 10 s.

Vytvořili jsme iluzi, že se auto pohybuje, protože scénérie před a za ním se pohybují a že scénérie v popředí se pohybuje rychleji, než v pozadí. Je to dáno tím, že předměty dále od nás se pohybují pomaleji než předměty blíže k nám.

Scény

Napravo od plátna se nám nachází seznam scén. Máme možnost tvořit nové scény (*Add Scene*), duplikovat aktuální scénu (*Duplicate Scene*) a mazat scénu (*Delele Scene*).

Na každou scénu si můžeme vytvořit jinou animaci. Po exportu animace se nám uloží animace s několika scénami. Výsledek vypadá jako film, kde je mezi scénami stříh.

3 VÝBĚR VHODNÉHO SOFTWARE PRO TVORBU ELEKTRONICKÉHO VÝUKOVÉHO KURZU

V této kapitole uvedu programy, které jsem použila k realizaci výukového kurzu. Hlavní část, tedy samotný kurz, jsem realizovala pomocí autorského systému ProAuthor. Multimediální komponenty (obrázky, animace a videa) jsem tvořila v programu Camtasia Studio 8 a ve vektorovém editoru Inkscape.

3.1 AUTORSKÝ SYSTÉM PROAUTHOR

ProAuthor je databázová aplikace určená pro tvorbu eLearningových kurzů. Byla vyvinuta společností Rentel a. s. a Západočeskou univerzitou v Plzni.

Aplikace je uživatelsky přívětivá a uživatel nemusí oplývat znalostmi pokročilého programování. Umožňuje přehledné členění jednotlivých kapitol, do kterých se vkládají studijní články, úkoly, cvičení, diskuse, autotesty, testy a ankety.



Obrázek 60: ProAuthor

Zdroj: ProAuthor [online]. [cit. 30.6.2016]. Dostupný na WWW: <http://athena.zcu.cz/ProAuthor/0.gif>

Hlavní částí kurzu jsou právě studijní články. Většinou, pokud nenastavíme jinou možnost, je okno rozděleno na dvě části, kde pravá strana obsahuje studijní text a na levé straně se nachází část s multimediálními komponentami. Do té lze vkládat tzv. MM&S komponenty (obrázky, tabulky, videa, animace, odkazy, soubor atd.). Do studijních článků můžeme přidávat i klíčová slova, která charakterizují pojmy obsažené ve studijních článcích.

Výsledný kurz je možné vyexportovat jako webové stránky (E-book), do Moodle a další.

Tento program jsem si pro tvorbu výukového kurzu zvolila z důvodu přívětivého uživatelského rozhraní, možnosti vkládat multimediální komponenty a z důvodu

dostupnosti studentské licence, kterou jsem získala od Katedry výpočetní a didaktické techniky. [22]

3.2 CAMTASIA STUDIO 8

Camtasia Studio je velmi šikovný program pro nahrávání dění na obrazovce počítače. Ovládání je velmi snadné, stačí si zvolit výřez části obrazovky, kterou chceme zaznamenávat a pak jen zmáčknout tlačítko pro nahrávání. Po nahrání si zvolíme možnost editace a na časové ose si výsledné video sestříháme a uložíme.

Je ale pod licencí shareware a proto jsem měla možnost vyzkoušet jej jen na 30 dní.

3.3 INKSCAPE

Inkscape je multiplatformní open-source vektorový editor. Jako svůj výchozí formát používá SVG. Program je podporován operačním systémem Microsoft Windows, Mac OS X a unixovými operačními systémy. Primární vývojovou platformou Inkscape je operační systém Linux. [23]

4 REALIZACE ELEKTRONICKÉHO VÝUKOVÉHO KURZU

V úvodu realizace elektronického výukového kurzu jsem si stanovila primární cíl kurzu, vytvořila si představu o budoucím kurzu a jeho funkci a ujasnila si cílovou skupinu uživatelů.

4.1 CÍL ELEKTRONICKÉHO VÝUKOVÉHO KURZU

Cílem elektronického výukového kurzu bylo vytvořit dostatečnou studijní oporu základů tvorby plošných vektorových animací určenou pro studenty pedagogické fakulty (jako jsem já), kterým se tvorba animací bude hodit pro budoucí účely vykonávání učitelské profese. Kurz ale není určen jen studentům pedagogické fakulty, může se hodit i stávajícím pedagogům nebo ostatním studentům, kteří se o problematiku animace zajímají.

Tvorba plošných vektorových animací	
Obsah	
1. Úvod: Tvorba plošných vektorových animací	
	1. Úvod: Tvorba plošných vektorových animací
2. Animace	
	1. Úvod: Animace
	2. Studijní článek: Definice
	3. Studijní článek: Princip
	4. Studijní článek: Fáze tvorby animace
3. SVG animace	
	1. Úvod: SVG animace
	2. Studijní článek: Formát SVG
	3. Studijní článek: Základní struktura SVG kódu
	4. Studijní článek: Společné vlastnosti elementů
	5. Studijní článek: Základní elementy
	6. Studijní článek: Další elementy
	7. Autotest: Autotest 1: Základní znalosti SVG
	8. Cvičení: Cvičení 1
	9. Studijní článek: SVG v XHTML a HTML5
	10. Studijní článek: SVG animace
	11. Studijní článek: Základní animační elementy
	12. Studijní článek: Atributy animací
	13. Studijní článek: Zápis kódu animací
	14. Studijní článek: Příklady zápisu animací
	15. Úkol: Úkol
	16. Autotest: Autotest 2: Znalost SVG animací
	17. Cvičení: Cvičení 2
4. Animace v editoru	

Obrázek 61: Ukázka výukového kurzu

4.2 ELEKTRONICKÝ VÝUKOVÝ KURZ

V úvodu každé kapitoly je napsáno úvodní slovo pro studenta s informací, co může od kapitoly, kterou se chystá studovat, očekávat. Doplněny jsou i cíle charakterizující znalosti, kterých by měl student po prostudování nabýt.

SVG animace

V této kapitole se seznámíš s formátem SVG, jeho základní strukturou a základními elementy zdrojového kódu.

Dále se seznámíš s vkládáním objektů zapsaných v SVG do zdrojového kódu webových stránek.

V druhé části této kapitoly získáš přehled o základních animačních elementech a jeho atributech a naučíš se zapisovat SVG kód tak, že si vytvoříš tvé první animace.

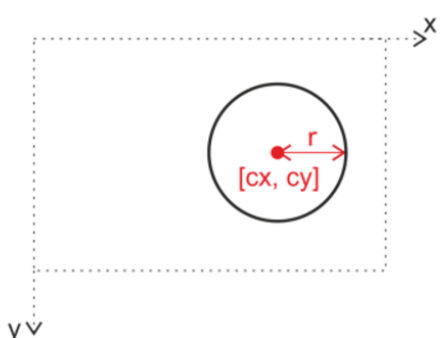
Cíle

Po přečtení této kapitole budeš schopen:

- popsat formát SVG
- popsat jeho hlavní výhody a nevýhody
- charakterizovat, jaké možnosti animace SVG má
- popsat možnosti vkládání SVG kódu do webových stránek
- charakterizovat základní elementy a jejich společné vlastnosti
- charakterizovat základní animační elementy a jejich atribuce
- zapsat jednoduchou animaci v SVG kódu

Obrázek 62: Ukázka úvodních informací výukového kurzu

Probíranou látku nalezne student ve studijních článcích, které jsou v nadpoloviční většině doplněny o názorné demonstrační obrázky, videa nebo animace. Student tak rovnou po prostudování daného článku na ukázkách vidí, jak co funguje, jak se co dělá a jak vypadá výsledek.

1	2	3	4	5	6	7	8
Kružnice							
<pre><circle stroke = "black" fill="none" cx="100" cy="50" r="30" /></pre> 							
<ul style="list-style-type: none"> • umožňuje vnořování elementu <code>svg</code> do jiných elementů 							
<p>line = čára, úsečka ¹ nejjednodušší geometrický tvar x_1, y_1 - souřadnice počátečního bodu x_2, y_2 - souřadnice koncového bodu</p>							
<p>rectangle = obdélník ² x_1, y_1 - souřadnice počátečního bodu <code>width</code> - šířka obdélníka <code>height</code> - výška obdélníka</p>							
<p>circle = kružnice ³ cx, cy - souřadnice středu kružnice <code>r</code> - poloměr</p>							
<p>ellipse = elipsa ⁴ cx, cy - souřadnice středu elipsy <code>rx</code> - poloměr osy x <code>ry</code> - poloměr osy y</p>							

Obrázek 63: Ukázka studijního článku

Kapitoly jsou doplněny i o pár autotestů, které ověřují základní znalosti po prostudování jednotlivých studijních článků.

AUTOTEST
<p>K čemu je určen formát SVG?</p> <ul style="list-style-type: none"> <input type="radio"/> k popisu dvourozměrné vektorové grafiky pomocí HTML <input type="radio"/> k zobrazení dvourozměrné bitmapové grafiky <input type="radio"/> k popisu dvourozměrné vektorové grafiky pomocí XML
<p>Co znamená pojme Scalable ve formátu SVG?</p> <ul style="list-style-type: none"> <input type="radio"/> při zvětšování a zmenšování obrázků ztrácí kvalitu <input type="radio"/> objekty v obrázku jsou matematicky popsané geometrické objekty <input type="radio"/> při zvětšování a zmenšování obrázků neztrácí kvalitu <input type="radio"/> možnost propojení s bitmapovou grafikou
<p>Jaká je nevýhoda formátu SVG?</p> <ul style="list-style-type: none"> <input type="radio"/> při zvětšování a zmenšování obrázků ztrácí kvalitu <input type="radio"/> formát neumožňuje propojení s bitmapovou grafikou <input type="radio"/> obrázky se složitěji pořizují <input type="radio"/> formát má omezený počet barev
<p>K pojmenování virtuálních souřadnic v základní struktuře kódu SVG se používá</p> <ul style="list-style-type: none"> <input type="radio"/> virtualBox <input type="radio"/> viewBox <input type="radio"/> virtualView
<p>Pokud u atributů neuvedeme jednotku, automaticky budou jednotky brány v</p> <ul style="list-style-type: none"> <input type="radio"/> milimetrech <input type="radio"/> centimetrech <input type="radio"/> pixelech <input type="radio"/> metrech

Obrázek 64: Ukázka autotestu

Kapitoly také obsahují cvičení, na kterých si student sám vyzkouší, jak se tvoří základní objekty ve zdrojovém kódu SVG a základní animace ve zdrojovém kódu SVG.

Další částí jsou také úkoly, které studenty průběžně učí práci s animačním editorem Animatron.

Animace v editoru
<p>Úkol 6</p>
<p>Zadání</p> <p>Vytvoř animaci demonstrující oběh planet okolo Slunce ve sluneční soustavě.</p>
<p>Tipy pro řešení</p> <p>Každá planeta se pohybuje jinak rychle a také rotuje kolem své osy.</p>
<p>▶ Pravidla hodnocení</p>

Obrázek 65: Ukázka zadání úkolu

ZÁVĚR

Cílem mé diplomové práce bylo vytvořit elektronický výukový kurz, který jeho uživatelům umožní proniknout do problematiky tvorby plošné vektorové animace, jejími principy, možnostmi vytváření animací a práce ve vybraném animačním editoru.

Kurz byl vytvořen za účelem vytvoření studijní opory tvorby plošných vektorových animací pro budoucí i stávající pedagogy a další studenty, kterým se tato dovednost bude hodit při vykonávání své učitelské nebo jiné profese.

Uživatelé by měli být schopni po absolvování kurzu orientovat se v problematice vektorové plošné animace a vytvářet jednoduché i složitější animace.

Vytvořený elektronický výukový kurz obsahuje 24 studijních článků rozdělených do 3 kapitol. Kurz je koncipován jako návod k vytváření animací od základů a je doplněn praktickými ukázkami tvoření vlastních animací.

Věřím, že jsem tuto diplomovou práci vypracovala dle zadaných požadavků a že v budoucnu poslouží jako vhodný výukový materiál pro stávající i budoucí pedagogy a studenty, kteří se chtějí naučit vytvářet vektorové plošné animace, ať už pro výuku nebo pro jejich osobní potřebu.

Samotný výukový kurz se nachází na přiloženém CD.

RESUMÉ

The thesis on the topic Implementation of e-learning course focused on creation of plane vector animations deals with the implementation of electronic training course.

Main part is focused on preparation of component (study articles, images, animation and video) training course.

Users should be able after the course understand the problems of plane vector animations and create simple and complex animations.

The course was designed to create a learning support plan creation vector animations for future and current teachers and other students, which will come in handy this skill in carrying out their teaching or other professions.

SEZNAM LITERATURY

- [1] **KALBAG, Asha.** *Počítačová grafika a animace*. 1. české vyd. Ilustrace Russell Punter. Bratislava: Příroda, 2001, 72 s. ISBN 80-070-0661-3.
- [2] **EISENBERG, J.** *SVG essentials*. 1st ed. Sebastopol, CA: O'Reilly, 2002, xvii, 335 p. ISBN 05-960-0223-8.
- [3] **JENKOV, Jakob.** *Tutorials: SVG Animation*. [online]. 2013 [cit. 2014-12-10]. Dostupné z: <http://tutorials.jenkov.com/svg/svg-animation.html>
- [4] **SUDA, Brian.** *Animating Vectors with SVG. 24 WAYS* [online]. 2013 [cit. 2014-12-10]. Dostupné z: <http://24ways.org/2013/animating-vectors-with-svg/>
- [5] **OPASCHOWSKI, Horst W.** *Einführung in die freizeit-kulturelle Breitenarbeit: Methoden u. Modelle d. Animation*. Bad Heilbrunn/Obb.: Klinkhardt, 1979. ISBN 3781503887.
- [6] *Prehistorie filmu* [online]. 2011 [cit. 2016-05-20]. Dostupné z: http://nemy-film.sweb.cz/Prehistorie_filmu/Uvod.html
- [7] *Obecná znalost formátu SWF* [online]. 2015 [cit. 2016-05-21]. Dostupné z: <http://www.apowersoft.cz/co-je-format-swf.html>
- [8] *W3C SVG Working Group* [online]. 2010 [cit. 2016-05-21]. Dostupné z: <https://www.w3.org/Graphics/SVG/>
- [9] **DOLEŽEL, Jan.** *Diplomová práce: Datové formáty pro prezentaci map na webu: Vektorové formáty* [online]. Praha, 2005 [cit. 2016-05-27]. Dostupné z: <http://geo3.fsv.cvut.cz/~soukup/dip/dolezel/index.html>
- [10] **KŘEPISH, Jiří.** *Animovaná tvorba #1: vznik a historie animace* [online]. 2015 [cit. 2016-05-29]. Dostupné z: <http://jirikrepish.cz/2015/12/23/animovana-tvorba-1-vznik-a-historie-animace/>
- [11] *2D Animation History* [online]. 2011 [cit. 2016-05-29]. Dostupné z: http://multimediamcc.com/old-students/ashaver/2d_history.html
- [12] *History of computer animation* [online]. [cit. 2016-05-29]. Dostupné z: https://en.wikipedia.org/wiki/History_of_computer_animation
- [13] *Synchronized Multimedia Integration Language (SMIL 3.0)* [online]. 2008 [cit. 2016-05-29]. Dostupné z: <https://www.w3.org/TR/REC-smil/>
- [14] *Extensible Markup Language (XML)* [online]. 2013 [cit. 2016-05-29]. Dostupné z: <https://www.w3.org/XML>

- [15] BOURRET, Ronald. Animatron Tutorial. [online]. , 33 s. [cit. 2016-06-20]. Dostupné z: <http://www.rpbourret.com/animatron/Bourret-AnimatronTutorial.pdf>
- [16] List of 2D animation software. Wikipedia, the free encyclopedia. [online]. 2014 [cit. 2016-06-15]. Dostupné z: http://en.wikipedia.org/wiki/List_of_2D_animation_software
- [17] *Main Page – Synfig Animation Studio*. [online] [cit. 2016-06-15]. Dostupné z: http://synfig.org/Main_Page
- [18] *Pencil2D – opensource animation software* [online]. [cit. 2016-06-15]. Dostupné z: <http://www.pencil2d.org/>
- [19] TOON BOOM ANIMATION INC. *2D Animation | Computer Animation Software | Toon Boom*. [online] 2008 [cit. 2016-06-15]. Dostupné z: <http://www.toonboom.com/products/toonBoomStudio/>
- [20] *Welcome Adobe Animate CC*. [online]. 2015 [cit. 2016-06-15]. Dostupné z: <http://blogs.adobe.com/animate/welcome-adobe-animate-cc-a-new-era-for-flash-professional/>
- [21] *Animatron: HTML5 Animation Editor* [online]. 2014 [cit. 2016-06-15]. Dostupné z: <https://blog.jetbrains.com/blog/2014/04/17/animatron-html5-animation-editor-inspired-by-middle-school-homework-assignment/>
- [22] KUBIŠTOVÁ, Zdenka a Zdeněk PEJSAR. *Tvorba e-learningových kurzů v prostředí ProAuthor* [online]. [cit. 2016-06-27]. Dostupné z: http://pokrok.ujep.cz/elektronicka_knihovna/Tvorba_e-learningovych_kurzu.pdf
- [23] KREJČÍ, Richard. GRAFIKA – Inkscape zblízka: Základní přiblížení *online+. 20.12.2004 [cit. 2014-05-12+]. Inkscape zblízka: Základní přiblížení. Dostupné z WWW: <<http://www.grafika.cz/rubriky/software/inkscape-zblizka-zakladni-priblizeni-132990cz.html>>.

SEZNAM OBRÁZKŮ

Obrázek 1: Phenakistoscope	5
Obrázek 2: Zoetrope	5
Obrázek 3: Ukázka elementu Line	25
Obrázek 4: Ukázka elementu Rectangle.....	25
Obrázek 5: Ukázka elementu Circle.....	26
Obrázek 6: Ukázka elementu Ellipse.....	27
Obrázek 7: Ukázka elementu Polyline	27
Obrázek 8: Ukázka elementu Polygon	28
Obrázek 9: Rodíl mezi elementem Polyline a Polygon.....	28
Obrázek 10: Rozdíl mezi výplní u elementu Polyline a Polygon.....	29
Obrázek 11: Ukázka rozdílu zápisu cesty.....	30
Obrázek 12: Ukázka rozdílu mezi zápoisem cest.....	31
Obrázek 13: Ukázka vložení SVG kódu přímo do zdrojového kódu webových stránek XHTML.....	32
Obrázek 14: Ukázka vložení SVG souboru do zdrojového kódu webových stránek.....	33
Obrázek 15: Ukázka rotace objektu.....	35
Obrázek 16: Ukázka transformace objektu	35
Obrázek 17: Ukázka změny měřítka objektu	36
Obrázek 18: Ukázka zkosení objektu	36
Obrázek 19: Počáteční stav elementu rect.....	40
Obrázek 20: Koncový stav elementu rect.....	40
Obrázek 21: Popis pracovní plochy Animatronu	45
Obrázek 22: Vlastnosti projektu	47
Obrázek 23: Ikona nástroje Výběr.....	48
Obrázek 24: Vlastnosti vybraného objektu	48
Obrázek 25: Ikona nástroje Přímý výběr	49
Obrázek 26: Ikona nástroje Pero	49
Obrázek 27: Rozšířená nabídka nástroje Pero	50
Obrázek 28: Ikona nástroje Tužka.....	50
Obrázek 29: Vlastnosti nástroje Tužka.....	50
Obrázek 30: Ikona nástroje Štětec	50
Obrázek 31: Uzavřená křivka	51
Obrázek 32: Rozšířená nabídka nástroje Štětec	51
Obrázek 33: Ukázka všech typů štětců.....	51
Obrázek 34: Ikona nástroje Text	52
Obrázek 35: Ikona nástroje Základní tvary	52
Obrázek 36: Rozšířená nabídka nástroje Základní tvary.....	52
Obrázek 37: Ukázka nastavení zaoblení nástroje Čtverec.....	52
Obrázek 38: Ikona nástroje Akce a události.....	53
Obrázek 39: Ikona nástroje Výplň.....	53
Obrázek 40: Ikona nástroje Guma	53
Obrázek 41: Rozšířená nabídka nástroje Guma	53
Obrázek 42: Ukázka nástroje Nůž.....	54
Obrázek 43: Ikona nástroje Kostra	54

Obrázek 44: Ukázka práce s nástrojem Kostra.....	55
Obrázek 45: Ikona nástroje Lupa.....	55
Obrázek 46: Ikona nástroje Ruka	55
Obrázek 47: Ikona Obchodu a knihovny	55
Obrázek 48: Ukázka nabídky Obchodu.....	56
Obrázek 49: Kombinace objektů	57
Obrázek 50: Popis časové osy	58
Obrázek 51: Ukázka přesunutí objektu	59
Obrázek 52: Ukázka úpravy cesty animovaného objektu	60
Obrázek 53: Ukázka umístění animovaného objektu na cestu	60
Obrázek 54: Ukázka nastavení řídicího bodu rotace	61
Obrázek 55: Ukázka časové osy při opakování části animace	62
Obrázek 56: Ukázka nastavení změny měřítka	62
Obrázek 57: Popis možností práce s vrstvami.....	63
Obrázek 58: Ukázka práce s vrstvami	64
Obrázek 59: Ukázka techniky Panning	65
Obrázek 60: ProAuthor.....	67
Obrázek 61: Ukázka výukového kurzu	69
Obrázek 62: Ukázka úvodních informací výukového kurzu	70
Obrázek 63: Ukázka studijního článku.....	70
Obrázek 64: Ukázka autotestu.....	71
Obrázek 65: Ukázka zadání úkolu.....	71

PŘÍLOHY NA PŘILOŽENÉM CD

- DP_Pribylova_Tereza.docx (text diplomové práce ve formátu .docx)
- DP_Pribylova_Tereza.pdf (text diplomové práce ve formátu .pdf)
- Tvorba plošných vektorových animací (E-book) (vyexportovaný kurz)
- Tvorba plošných vektorových animací (ProAuthor)