

MACROMEDIA FLASH A JEHO VYUŽITÍ PŘI VÝUCE TECHNICKÝCH PŘEDMĚTŮ

Martin Draessler, Jan Jirsa, Martin Molhanec

Katedra elektrotechnologie, Fakulta elektrotechnická, České vysoké učení technické v Praze
Technická 2, 166 27 Praha 6, Dejvice
E-mail: draessm@fel.cvut.cz, jirsaj@fel.cvut.cz, molhanec@fel.cvut.cz

Abstrakt

Cílem této práce je seznámit čtenáře s možnostmi programování v produktu Macromedia Flash. Vzhledem k tomu, že většina uživatelů Internetu zná Flash spíše po grafické stránce, zaměříme se v tomto příspěvku především na jeho podporu objektových technik, vnitřní skriptovací jazyk (tzv. ActionScript) a návaznost na grafické prvky.

1. Úvod

V průběhu posledních několika let došlo díky novým technologiím k výrazným změnám zejména v oblasti prezentace různých dat na Internetu. Již se neomezujeme na pouhé texty a obrázky, ale můžeme plně využít některou z moderních metod prezentací. K nim bezesporu patří streamované audio i video a samozřejmě také nesmíme opomenout nejrůznější animované prvky, se kterými se denně setkáváme ve formě reklam a bannerů. Většina těchto animací bývá vytvořena v produktu Macromedia FLASH. Byť by se mohlo zdát, že tento produkt je využíván hlavně pro reklamní účely, není tomu tak. V našem případě používáme FLASH pro zlepšení výuky, kdy pomocí animací zobrazujeme nejrůznější technologické procesy a výrobní postupy. FLASH lze však použít i pro tvorbu webových aplikací nejrůznějšího charakteru.

Vzhledem k tomu, že FLASH není jen vektorový grafický editor, ale zahrnuje v sobě i skriptovací jazyk ActionScript, zaměříme se právě na tuto jeho část. Budeme se zabývat problematikou objektů, jejich tvorbou a využitím v návaznosti na animované prvky. V závěru si na příkladu ukážeme, jak vytvořit interaktivní animaci *Snellova zákona*, týkající se odrazu a lomu světla.

2. ActionScript

Tento skriptovací jazyk se podobá do jisté míry JavaScriptu. V samotné animaci pak přispívá k lepší manipulaci s grafickými objekty a umožňuje tak vytvářet různá interaktivní prostředí. Nezanedbatelným efektem je i menší výsledná velikost animace, která samozřejmě souvisí s rychlostí jejího načítání do prezentované webové stránky.

Na úvod je třeba sdělit, že ActionScript (podobně jako JavaScript) je objektově orientovaný skriptovací jazyk. Setkáme se tedy s termíny, jako je např. *třída, objekt, instance, metoda* atd.

2.1. Předdefinované třídy

Skriptovací jazyk ActionScript nám nabízí již několik předdefinovaných tříd. Jedná se o tyto třídy:

Array	- Třída určená pro práci s poli.
Boolean	- Jednoduchá třída pro práci s logickými proměnnými.
Color	- Umožňuje získávat, nastavovat a transformovat hodnoty RGB u klipů.
Date	- Pomocí této třídy lze získat datum a čas ve vztahu k UCT nebo času operačního systému.
Key	- Třída pro snímání vstupu z klávesnice.
Math	- Třída pro práci s matematickými funkcemi.
MovieClip	- Třída pro práci s animacemi.
Number	- Jednoduchá třída pro práci s daty číselného typu.
Object	- Jedná se o kořenovou třídu v hierarchii tříd ActionScriptu. Pomocí ní můžeme vytvářet vlastní objekty.
Selection	- Třída, která zajišťuje nastavení a řízení aktuálních editovatelných textových polí.
Sound	- Umožňuje nastavit a řídit zvuky v jednotlivých instancích klipů nebo globálně pro celou animaci.
String	- Třída určená pro práci s řetězci.
XML	- Třída určená pro práci s XML.
XMLSocket	- Třída, která implementuje klientské sockets pro komunikaci se serverem.

Každá z výše zmíněných tříd pak samozřejmě disponuje řadou metod a atributů, jež dovolují vytvořený objekt efektivně využívat. Seznam těchto metod je možné nalézt např. v [1].

2.2. Datové typy

Kromě předdefinovaných tříd je třeba se ještě zmínit o datových typech. Existují dva základní typy, které ActionScript poskytuje:

- primitivní datové typy
- referenční datové typy

Do první skupiny jsou zařazeny typy: *řetězec (String)*, *číslo (Number)* a *logický typ (Boolean)*, do druhé pak *klip (Movie clip)* a *objekt (Object)*. K těmto datovým typům je nutné říci několik poznámek. První se týká čísel a jejich reprezentace. Ta jsou zpracovávány v plovoucí řádové čárce s dvojitou přesností. Pro operace s nimi lze použít jak aritmetické operátory, tak i metody z třídy Math. Druhá poznámka se týká animovaných klipů. Tento typ je v podstatě referencí na grafický element, který reprezentuje animaci na ploše. Pro operace s tímto typem je možné použít některou z metod třídy MovieClip.

2.3. Tvorba objektů

Nyní se podíváme, jak vytvoříme vlastní objekt nebo objekt na základě předdefinovaných tříd. K tomu využijeme dvou operátorů. Prvním z nich je new, který slouží k tvorbě instance třídy založené na předdefinované třídě, případně na objektu definovaném uživatelem. Například objekt třídy Date vytvoříme následujícím způsobem:

```
dnesni_datum = new Date();
```

Pokud bychom chtěli získat např. pořadí dne v týdnu, stačí použít metodu getDay. Příkaz bude vypadat následovně:

```
cislo_dne = dnesni_datum.getDay();
```

Podobný je i přístup k atributům dané instance. Pomocí operátoru new můžeme však také vytvářet nové objekty a to s pomocí námi definované funkce (*konstrukturu*). Opět si ukážeme příklad:

```
function Ctverec(strana) {  
    this.strana = strana;  
    this.plocha = strana*strana;  
}
```

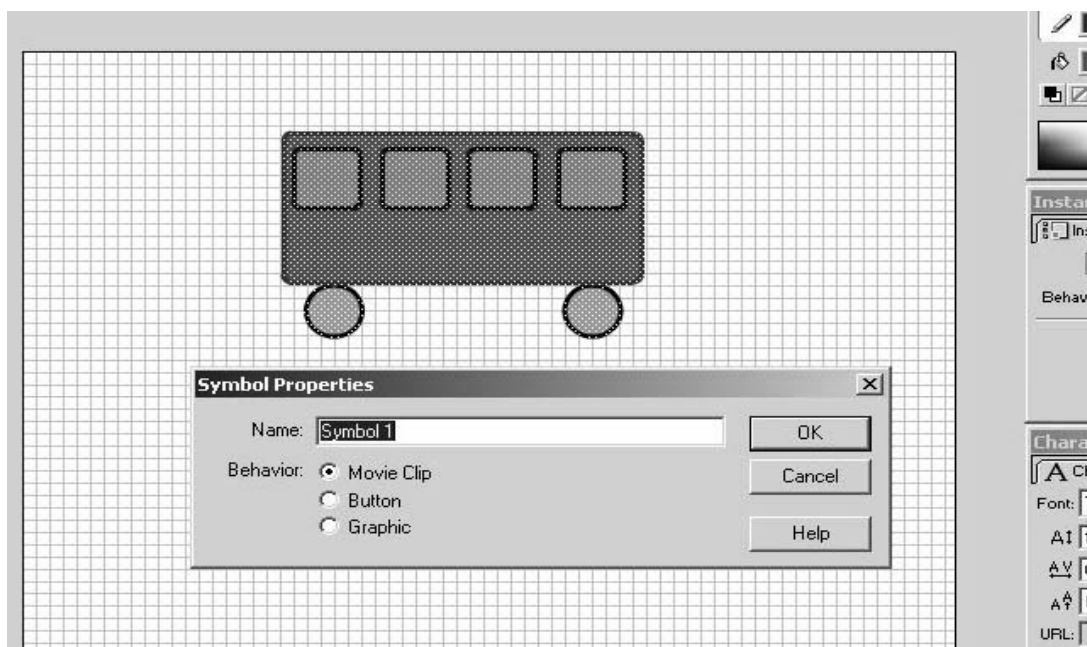
```
var Muj_Ctverec = new Ctverec(4);
```

Vraťme se nyní k druhé možnosti a to k operátoru inicializace {}. Ten nám umožňuje přímo vytvořit nový objekt a současně ihned nastavit hodnoty jeho proměnných. Předchozí příklad si přepíšeme pomocí operátoru inicializace takto:

```
Muj_Ctverec = {strana: 5, plocha:(strana*strana)};
```

Tímto způsobem je možné vytvářet nové objekty v ActionScriptu, které nejsou instancí některé z předdefinovaných tříd (resp. jsou instancí třídy Object). Je zřejmé, že funkce konstruktor, nám dovoluje vytvářet jakési „*vlastní třídy*“. Grafické objekty je však možné vytvářet přímo v grafickém editoru. Postup si popíšeme v následujících krocích:

- Nakreslíme grafický element
- Provedeme konverzi do požadovaného typu objektu (Obr. 1)
Máme na výběr ze třech možností – klipu, tlačítka nebo obrázku.
- Nyní jsme získali objekt do knihovny a současně máme též instanci tohoto objektu na ploše.



Obr. 1 Ukázka tvorby objektu

Tento způsob vytváření grafických objektů je pro většinu uživatelů asi nejobvyklejší. S takto vytvořeným objektem můžeme provádět nejrůznější akce podle toho, pro jaký má účel byl vytvořen.

3. Příklad animace Snellova zákona

Na jednoduchém příkladu si ukážeme, jakým způsobem je možné vytvořit interaktivní animaci za pomoci ActionScriptu. Nebudeme se zde zabývat grafickou stránkou problému, spíše budeme klást důraz na programovou část řešené úlohy.

Na úvod je třeba říci něco k zadání: Budeme chtít vytvořit model Snellova zákona, tak aby bylo možné nastavovat úhel paprsku dopadajícího na rozhraní dvou prostředí s různým indexem lomu a současně, aby bylo možné sledovat, jak se mění úhly odraženého a procházejícího paprsku. Nastavování úhlu dopadu bude realizováno posuvníkem a hodnotu úhlu budeme zobrazovat v číselné podobě. Při realizaci nesmíme zapomenout na skutečnost, že úhly se měří vždy od kolmice vedené na rozhraní obou prostředí. Ovládání modelu uživatelem bude rozděleno na dva po sobě následující kroky. V prvním z nich nastavíme posuvníkem požadovaný úhel dopadu. V druhém stiskneme tlačítko, které symbolizuje aktivování zdroje světla a tím dojde k zobrazení paprsků. Poté je i nadále možné měnit úhel dopadajícího paprsku.

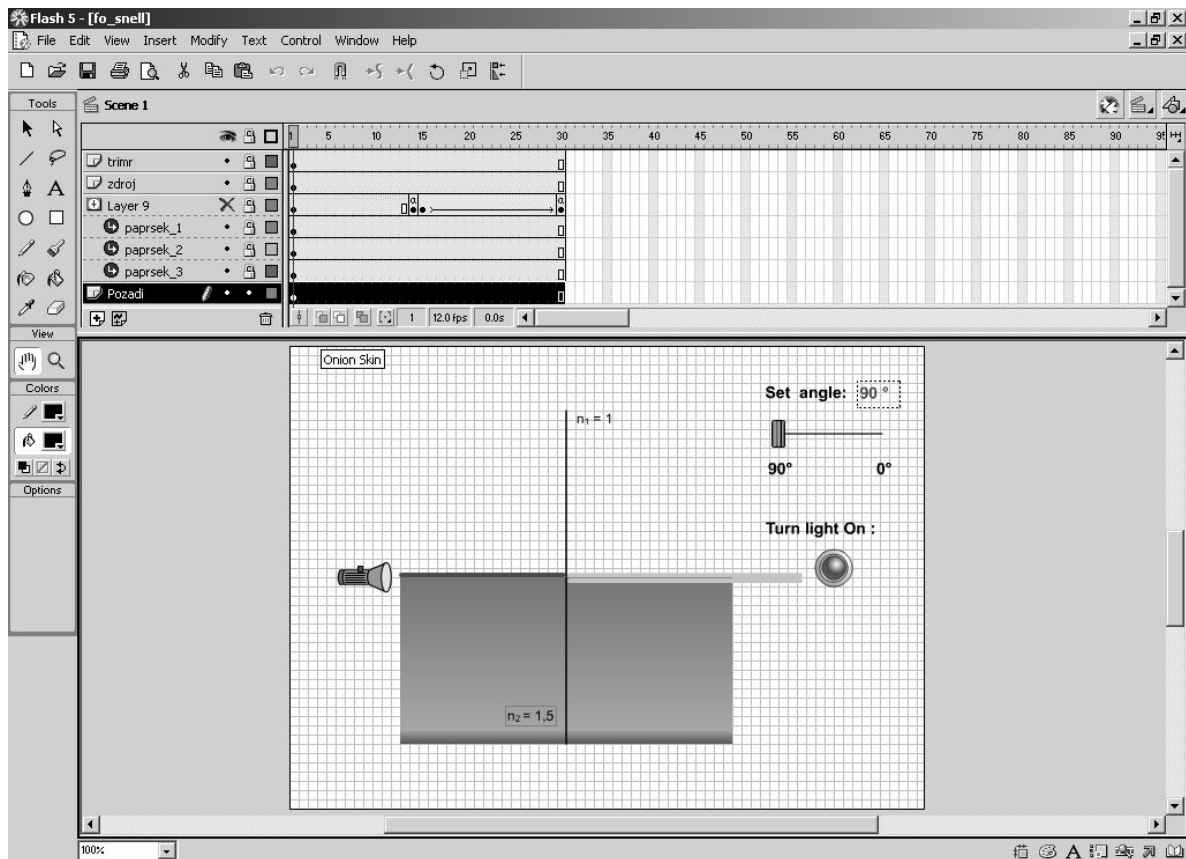
Nyní již k samotné realizaci celé animace, kterou si rozdělíme do několika fází.

- Nejprve nakreslíme všechny statické prvky, tzn. samotné rozhraní a popisky.
- Dále postupně vytvoříme grafické objekty, jež budou reprezentovat zdroj světla, paprsky, posuvník a tlačítko (Obr. 2).
- V dalším kroku pojmenujeme jednotlivé instance objektů (Obr. 3).
- V tomto okamžiku máme vše připraveno pro tvorbu samotného skriptu, jenž bude zajišťovat samotnou funkčnost modelu. Tento skript bude vypadat následovně (Obr. 4):

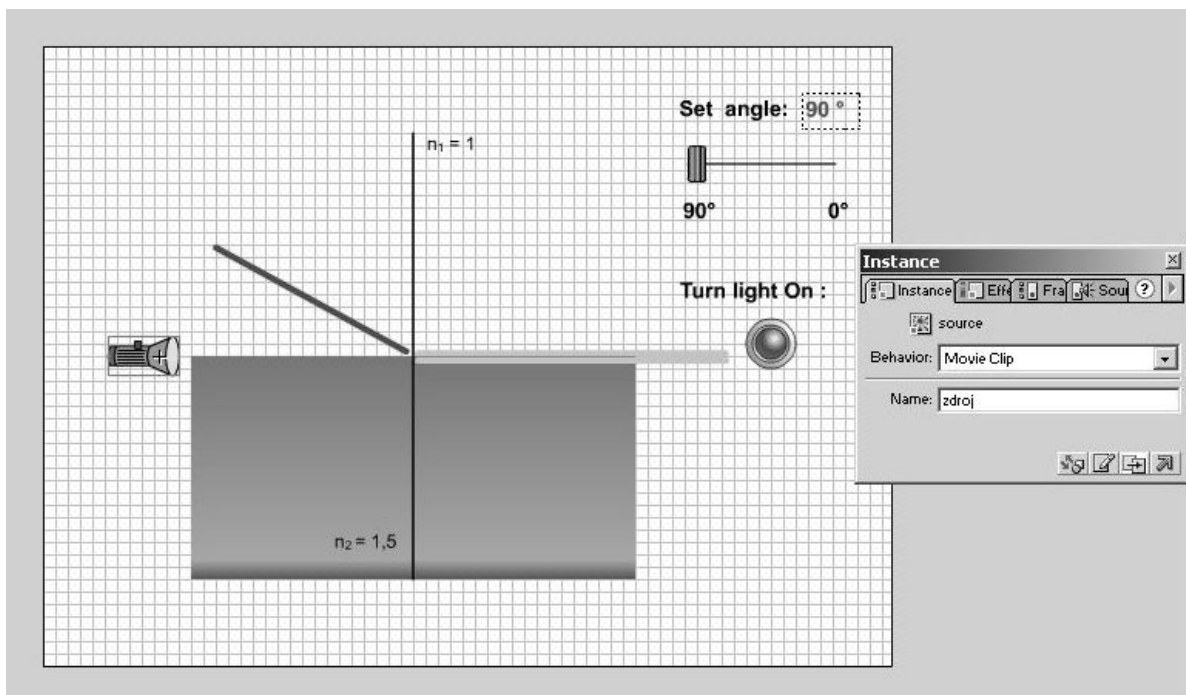
```
onClipEvent (load) {  
    top = _y;  
    bottom = _y;  
    left = _x;  
    right = _x+90;  
    pom = _x;  
    zd_x = _root.zdroj._x;  
    zd_y = _root.zdroj._y;  
    _root.paprsek2._rotation = 48;  
}
```

```
onClipEvent (enterFrame) {  
    if (tah == true) {  
        uhel = Math.round(_x-pom);  
        uhel_s = 90-uhel;  
        _root.ang = uhel_s + " °";  
        _root.zdroj._rotation = uhel;  
        _root.zdroj._x = zd_x-165*Math.cos((Math.PI/180)*uhel)+165;  
        _root.zdroj._y = zd_y-165*Math.sin((Math.PI/180)*uhel);  
        _root.paprsek1._rotation = uhel;  
        _root.paprsek2._rotation = 90-  
            Math.round((180/Math.PI)*Math.asin(1/1.5*Math.sin((Math.PI/180)*uhel_s)));  
        _root.paprsek3._rotation = -uhel;  
    }  
}
```

e) Nyní již zbývá animaci odladit a vyexportovat tak, aby ji bylo možné začlenit do webové stránky (Obr. 5).



Obr. 2 Základní grafické rozvržení



Obr. 3 Pojmenování instancí

The screenshot shows a software environment for creating a Snell's Law model. The workspace is a grid with a light source (flashlight) on the left, a screen in the center, and a detector on the right. A control panel at the top left shows $n_1 = 1$ and a slider for 'Set angle: 90°'. The 'Object Actions' panel is open, displaying the following JavaScript code:

```

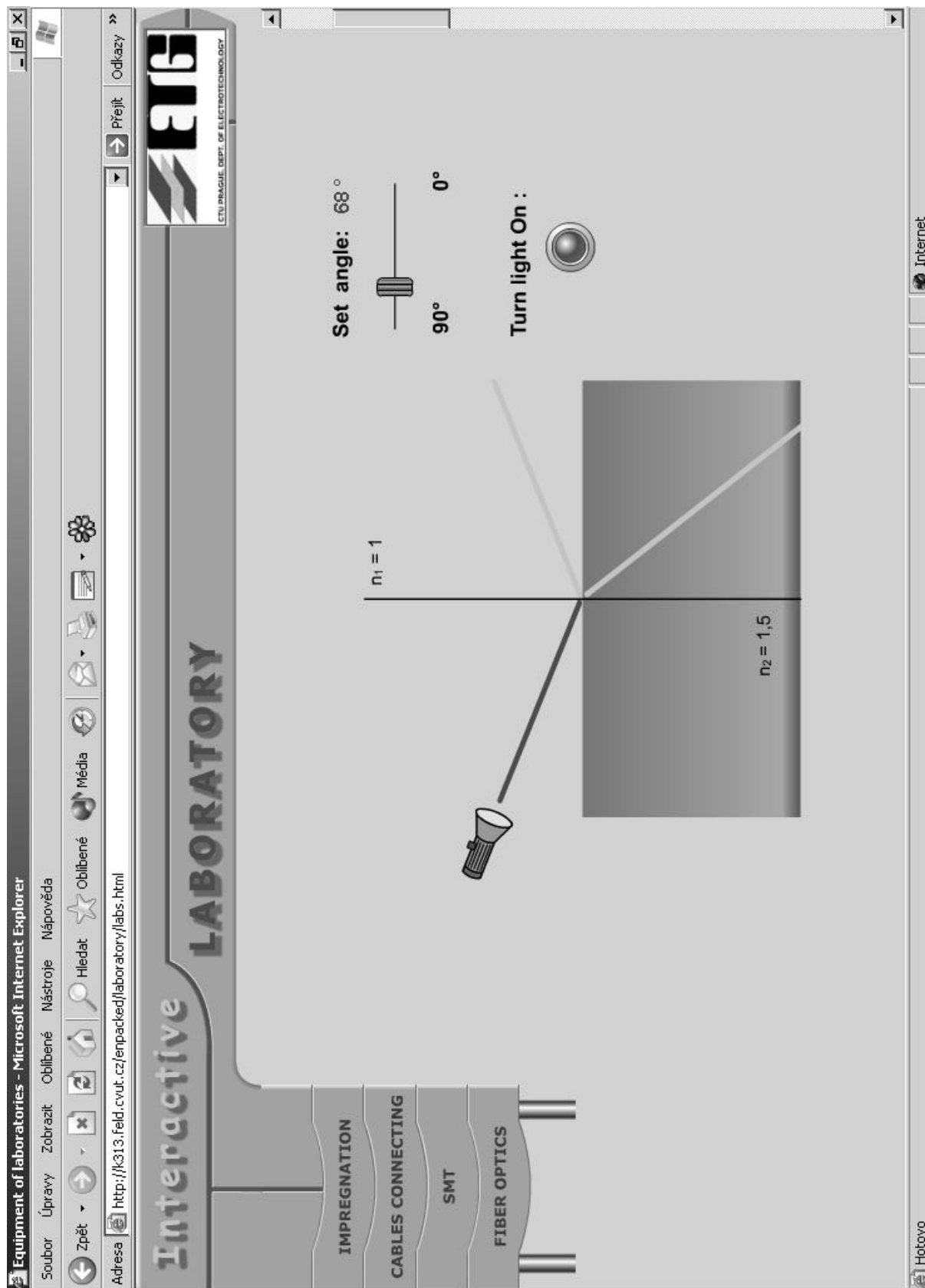
onClipEvent (load) {
    top = _y;
    bottom = _y;
    left = _x;
    right = _x+90;
    pom = _x;
    zd_x = _root.zdroj._x;
    zd_y = _root.zdroj._y;
    _root.paprsek2._rotation = 48;
}

onClipEvent (enterFrame) {
    if (tah == true) {
        uhel = Math.round(_x-pom);
        uhel_s = 90-uhel;
        _root.ang = uhel_s + " °";
        _root.zdroj._rotation = uhel;
        _root.zdroj._x = zd_x-165*Math.cos((Math.PI/180)*uhel)+165;
        _root.zdroj._y = zd_y-165*Math.sin((Math.PI/180)*uhel);
        _root.paprsek1._rotation = uhel;
        _root.paprsek2._rotation = 90-Math.round((180/Math.PI)*Math.asin(1/1.5*Math.sin((Math.PI/180)*uhel_s)));
        _root.paprsek3._rotation = -uhel;
    }
}

```

The status bar at the bottom right indicates 'Line 24 of 24, Col 1'.

Obr. 4 Skript pro vytvoření modelu Snellova zákona



Obr. 5 Konečná podoba webové stránky

4. Závěr

Cílem příspěvku bylo seznámit čtenáře s možnostmi programování a tvorby interaktivních animací v produktu Macromedia FLASH. Je možné říci, že ačkoliv se ActionScript nedá svými možnostmi srovnávat např. s C++ nebo podobnými jazyky, je jeho vybavení pro účely, které mu jeho tvůrci předurčili, zcela dostačující. Zvláště pokud vezmeme v úvahu, že ActionScript slouží především jako užitečné rozšíření možností práce a manipulace s grafickými objekty.

Pokud se podíváme na vývojový nástroj FLASH podrobněji, zjistíme, že je důsledně objektově orientován. V okamžiku, kdy vytvoříme libovolný klip, automaticky se z něj stává objekt, který má sadu atributů a metod, jež můžeme využívat.

Na závěr je možné konstatovat, že samotný Flash je vhodným nástrojem nikoli pouze pro pobavení návštěvníků některých webových sídel, ale i nástrojem pro seriózní podporu výuky různých oborů. Autoři tohoto příspěvku využívají FLASH pro podporu výuky technologických procesů a fyzikálních jevů. Přes poměrně krátkou dobu využití tohoto prostředku (2 roky) se zdá, že uplatnění animací ve výuce je značným přínosem a zlepšením celého pedagogického procesu.

Využitím nástroje Macromedia FLASH ve výuce technologických procesů se naše pracovní skupina na katedře elektrotechnologie ČVUT-FEL zabývá již delší dobu. Flashové animace jsou součástí projektu ENPACKED jehož cílem je vybudovat webové sídlo věnované technologickým procesům. Výsledky naší předchozí práce byly publikovány v rámci několika, zejména technologických, konferencí [3], [4], [6], [7] a [8] a byly také obsahem jedné diplomové práce [5]. Konkrétní animace je možné si prohlédnout na adrese [2], kde jsou umístěny webové stránky (*Virtuální laboratoř*), které jsou momentálně v testovacím provozu.

Literatura:

1. ActionScript Reference Guide, Macromedia FLASH
2. Virtuální laboratoř - <http://k313.feld.cvut.cz/enpacked/laboratory/labs.html>
3. Jirsa, J., „Elektronická výuka v rámci projektu ENPACKED“, sborník 1.ročníku soutěže o nejlepší vědeckotechnický projekt, vyd. ASTRA, Praha 2000
4. Jirsa, J., „Objekty a Macromedia FLASH“, poster (nebylo ve sborníku publikováno), Objekty 2002, ČZU, Praha 2002
5. Jirsa, J., „Kabely pro elektrotechniku – Virtuální laboratoř“, diplomová práce, ČVUT-FEL, katedra elektrotechnologie, Praha 2003
6. Molhanec, M., “Networking the Electronics Packaging Education”, In: ISSE 2000 Conference Proceeding. Budapest: Technical University, 2000, vol. 1, p. 75-78. ISBN 963-420-639-5.
7. Molhanec, M., „Using of Macromedia FLASH for Development of Internet Based Lectures at the Department of Electrotechnology of the CTU Prague“. In: ISSE2001. Bucharest: Politehnica, 2001, vol. 1, p. 107-111. ISBN 0-7803-7111-9.
8. Molhanec, M., „Using of Macromedia FLASH for Development of Internet Based lectures at the Department of Electrotechnology of the CTU Prague“. In: 5th International Academic Conference on Electronic Packaging Education and Training. Templin/Uckermark: Verlag Dr. Markus A. Detert, 2002, p.196-202. ISBN 3-934142-07-9.