

DUM č. 1 v sadě

28. Inf-4 Jednoduchá hra Had ve Flashi (ActionScript)

Autor: Robert Havlásek

Datum: 26.02.2013

Ročník: 5AV

Anotace DUMu: Flash - teorie: Principy událostmi řízeného programování. Reakce na událost, obsluha události (klik myší, stisk klávesy). Rozbor kláves, na které lze reagovat. Událost EnterFrame.\n

Materiály jsou určeny pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tvorba hry Had v ActionScriptu ve Flashi

Úvod do sady

Následující sada 20 DUMů popisuje, jak studenty naučit teorii, procvičit ji na příkladech a implementovat tak, aby celá skupina zvládla vyrobit jednoduchou hru Had (podle vzoru). Studenti vesměs vědí, o jakou hru jde – stačí vyslovit „Had na Nokii“ – v roce 1998 začala jednu z alternativ této hry do svých mobilních telefonů předinstalovávat firma Nokia.

Každá dvojice DUMů (liché+sudé číslo) tvoří ucelený blok – doporučuji učit je jako dvojhodinovku: v lichém DUMu probereme teorii a vyrobíme samostatný krátký příklad bez návaznosti na hru Had, v sudém DUMu implementujeme získané poznatky do tvořené hry. Součástí každého sudého DUMu bude kompletní výpis zdrojového kódu, který jsme dosud napsali.

Popis hry

Kurzorovými šipkami řídíte pohyb hlavy hada, za níž se táhne jeho tělo. Cílem je sbírat objevující se jablka a nenajet do zdi ani do svého těla. Při sebrání jablka se tělo o jeden díl prodlouží, je možné též implementovat různé levely (úrovně obtížnosti) s různou rychlostí pohybu či různě rozmístěnými zdmi.

Programovací prostředí

Hru i teorii k ní vyučujeme v prostředí Adobe (dříve Macromedia) Flash a v jeho jazyce ActionScript. Používáme verzi Macromedia Flash 8.0 a ActionScript 2.0, oboje z roku 2005, kdy jsme jako škola pořídili multilicenci balíku Macromedia Studio 8. Všechny uváděné skripty (algoritmy i syntaxe) jsou plně kompatibilní i s vyššími verzemi (až do aktuální CS6 a ActionScriptu 3.0).

K ActionScriptu lze získat mnoho výukových materiálů (knih, online kurzů, ...), pro oficiální dokumentaci doporučuji web <http://livedocs.adobe.com/flash/9.0/main/>. Vhodnou nápovědu lze získat i přímo v programu (stisknutím klávesy F1 ve vhodném místě, např. při psaní kódu).

Chcete-li si o ActionScriptu číst česky, kupte si knihu „ActionScript 3.0 oficiální výukový kurz“ nakladatelství Computer Press (ISBN 978-8-02-513335-4). Případně lze základní dovednosti získat ze seriálu na webu <http://programujte.com/clanky/36-serial-actionscript/> nebo na <http://www.havrlant.net/flash/seznameni.php>, zajímavé tipy a zpracované ukázky najdete i na <http://www.flash.cz/portal/clanky.aspx?sekce=36>

Předpoklady kladené na studenty

Obvykle před výukou této sady následuje výuka práce ve Flashi (DUMy 10. až 20. sady Animace), studenti by se měli umět v programu pohybovat, vyrobit základní kresbu, převést ji do symbolu, pracovat s časovou osou, pracovat s knihovnou.

Rozvoj abstraktního myšlení přichází obvykle též s věkem, programovat v ActionScriptu v celé skupině doporučuji v předmaturitním nebo maturitním ročníku, s mladšími studenty výjimečně.

Principy událostmi řízeného programování

Existuje několik programovacích paradigmat (způsobů, jak k programu jako celku přistoupit). Člověku je nejbližší tzv. strukturované programování, tedy zápis celého programu „shora dolů“, dá se chápat jako „pracovní postup“, ideální pro nějaký výpočet bez zásahu uživatele (resp. nanejvýš se zadáváním hodnot).

Pomineme-li poněkud extravagantní funkcionální programování a logické programování, zůstane programování událostmi řízené. Program samostatně provádí pouze drobné úkoly a čeká na uživatelské zadání (tzv. událost), na které poté reaguje spuštěním nějakého kusu programu (tzv. obsluhou události).

Událost může vyprodukovat i část systému, jiný program v systému nebo jiný počítač (např. tiknutí hodin časovače, výzva k uzavření okna, příchod dat po síti, ...)

Reagovat může i víc programů (představme si program, který sedí u hodin v systrayi a počítá kliknutí myši – reaguje vždy při každém kliknutí kdekoliv ve Windows), v daném programu může na jednu událost reagovat i víc objektů.

Seznam běžných událostí ve Flashi

Symbol typu Graphic není pro události až tak vhodný, nejlépe se ve Flashi programují události v symbolu typu MovieClip či Button. Může v nich nastat tato běžná událost:

press	Uživatel na objektu stisknul levé tlačítko myši (směrem dolů)
release	Uživatel na objektu pustil levé tlačítko myši (směrem nahoru)
releaseOutside	Uživatel pustil levé tlačítko myši mimo objekt (když jej předtím stisknul na objektu – typicky se snaží objekt „přetáhnout“)
rollover	Myši jedeme přes objekt (aniž bychom tlačítko myši mačkali)
rollOut	Myši jsme právě odjeli z objektu
dragOver	Uživatel kdysi předtím nějaký jiný objekt chytil a nyní s tím přetahuje přes „naš“ objekt
dragOut	Uživatel už nepřetahuje nad „naším“ objektem, odjel jinam
keyPress("kód_klávesy")	Uživatel napsal něco na klávesnici (do uvozovek je nutné napsat speciální kód stisknutého tlačítka, viz později)

U celé animace a též u každého MovieClipu mohou nastat ještě tyto klipové události:


load	MovieClip se právě načel do paměti a začíná se spouštět
enterFrame	Animace přeskočila na další snímek v časové ose – tato událost nastává poměrně často (přesně tak často, jaké je fps animace)
mouseMove	Uživatel s myší pohybuje (kdekoliv, nejen nad MovieClipem)
mouseDown	Uživatel stisknul levé tl. myši (kdekoliv, nejen nad MovieClipem)
mouseUp	Uživatel stisknul levé tl. myši (kdekoliv, nejen nad MovieClipem)
keyDown	Uživatel stisknul klávesu na klávesnici (směrem dolů)
keyUp	Uživatel uvolnil klávesu na klávesnici (směrem nahoru)

Pedagogická poznámka: Událostí je víc, některé studentům zatajíme. I tento výčet je poměrně dlouhý, ne až tak úplně zajímavý, obvykle jej dám studentům na papíře a rychle je proběhnu s odkazem na později.

Naše první událost a reakce na ni

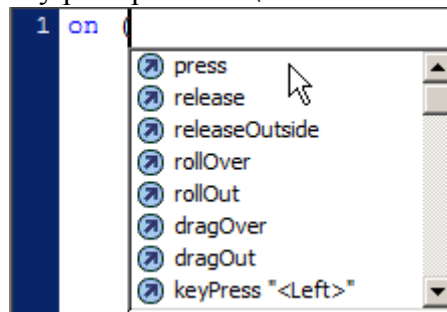
Spustíme Flash s novým dokumentem. Do plochy nakreslíme čtverec (bez okraje, pouze s výplní), převedeme jej do symbolu typu MovieClip, jméno může zůstat Symbol1. Označíme jej. *Pedagogická poznámka: Studenti mívají tendenci na označení objektu zapomenout, píší potom kód do Actions plochy, ne do Actions čtverce. Je vhodné je na tento fakt důrazně upozornit.*

Klikneme do panelu Actions, v záložce vlevo dole pod bílým místem se ujistíme, že jde

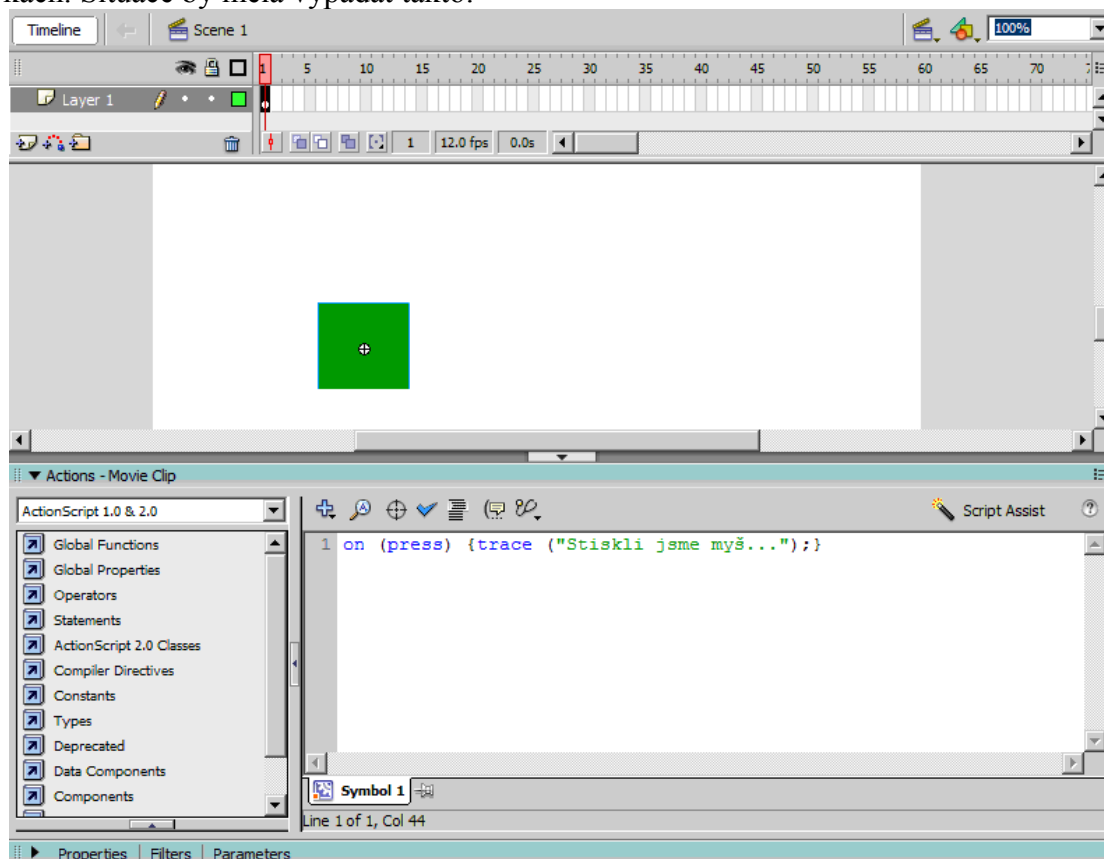
o Actions čtverce (je tam záložka  Symbol 1 ) a do bílého místa vpravo napíšeme kód:

```
on (press) {trace ("Stiskli jsme myš...");}
```

Pokud máme při psaní zapnuté tlačítko  CodeHint (=našeptávač), bude nám Flash v některých situacích napovídat, typicky po zapsání on (:



Povšimneme si též různých barev znaků – modrá znamená klíčové slovo, zelená text v uvozkách. Situace by měla vypadat takto:

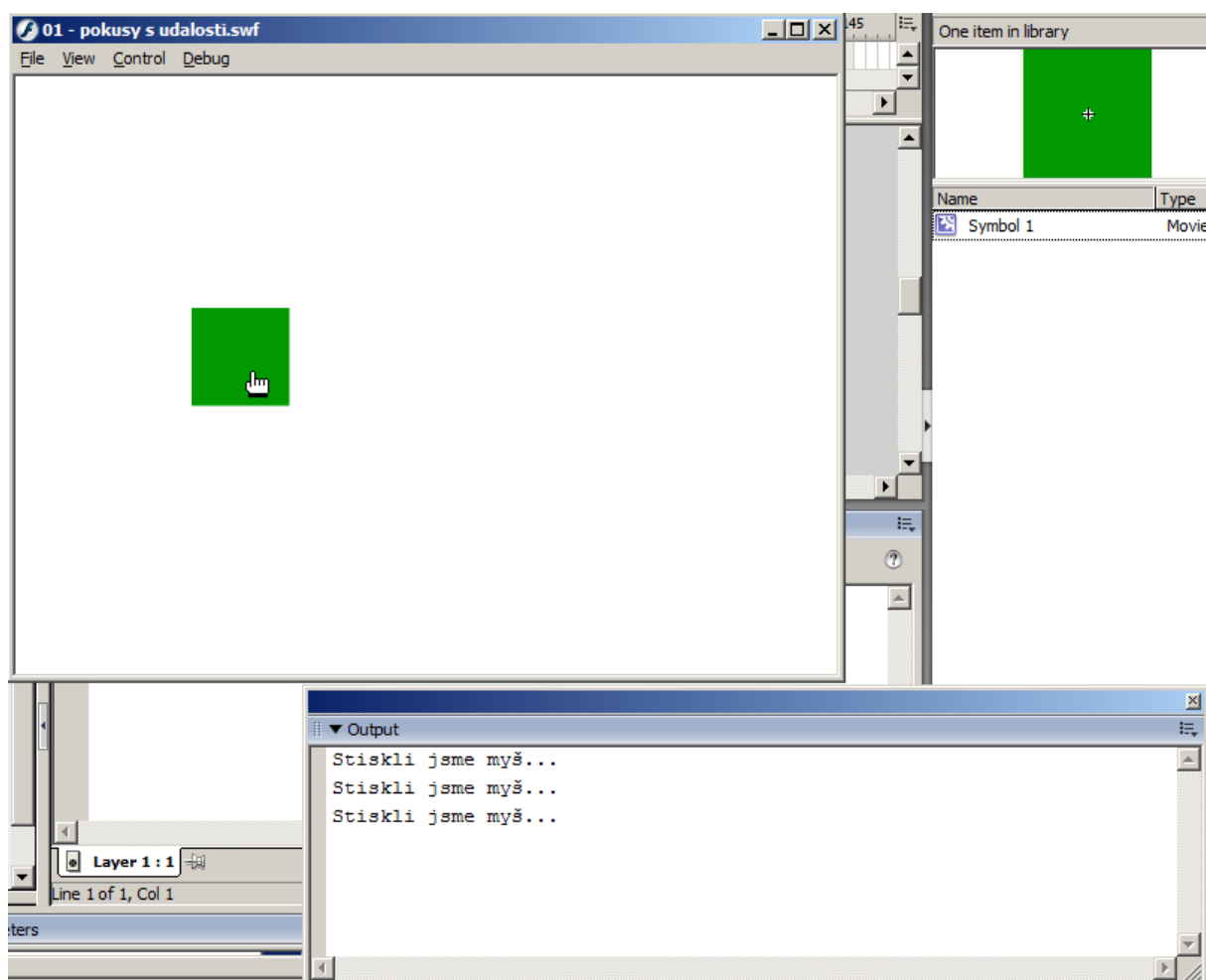


Příkaz `trace` je vhodný při ladění programu – vypisuje text do speciálního okna Output; programátor tak může sledovat, kdy co proběhlo.

Animaci uložíme, spustíme stiskem CTRL+Enter a vyzkoušíme, že po najetí na čtverec kurzor změni tvar (na ručičku značící „můžeš na mě kliknout“), klikneme-li, objeví se flashové okno Output a v něm text „Stiskli jsme myš...“

Vyzkoušíme, kdy přesně se nápis objeví; co se stane, když myší nejen klikneme, ale stiskneme tlačítko a chvíli počkáme, když stiskneme tlačítko a odjedeme jinam, když stiskneme pravé tlačítko myši, prostřední tlačítko myši, zarolujeme kolečkem myši...

Správný programátor se pozná podle toho, že experimentuje („...a jak ten program zareaguje, když...“)



Pedagogická poznámka: Visí nám tu najednou poměrně hodně oken, jednak ve Flashovém programu, pak okno animace a vedle ještě okno Output. Občas je tato situace horší na orientaci. V okně „Output“ je vlastně jen Flashový panel, ten lze umístit (Drag&Drop za horní pruh) i dovnitř do okna Flashe, třeba pod knihovnu Library – podrobně přesuny panelů popisují na 2. straně 10. DUMu animační sady...

Obvykle studentům ukazují společně, že spustím-li vyrobený soubor .SWF doklikáním přes Tento počítač, ... v externím přehrávači (mimo Flashový program), okno Output neuvídím. To proto, aby si pomocí příkazu `trace` nechtěli „povídat s uživatelem“ – pro tuto komunikaci jsou nástroje typu dynamický text (viz DUM č. 17 této sady).

Reakce na stisk klávesy

Stisk klávesy se vždy týká všech objektů v animaci. Obsluhu události lze psát do Actions libovolného objektu typu MovieClip, na událost tak může reagovat víc objektů zároveň. Rozlišujeme lokálně obsluhovaný `on(KeyPress)`, který reaguje na „napsatelný“ znak, a globální `onClipEvent(KeyDown)` či `onClipEvent(KeyUp)`, jež reagují na stlačení či uvolnění libovolné klávesy na klávesnici, včetně systémových kláves jako Shift, CapsLock, Ctrl, atp. Do Actions našeho čtverce napíšeme kód:

```
onClipEvent(KeyDown) {trace (Key.getCode());}
```

Tento kód způsobí, že kdykoliv je stlačena nějaká klávesa, její číslo se nám vytrasuje do okna Output.

Pedagogická poznámka: Studenty vyzveme ke stisku CTRL+Enter a necháme je experimentovat; položíme např. otázku „Které klávesy nefungují a proč?“ Obvykle též vyrobím dynamický text na ploše a nechám do něj zobrazovat kód stisknuté klávesy, abychom vyzkoušeli, které klávesy fungují při spuštění .SWF souboru v externím přehrávači (kde okno Output není). Teorie, jak vyrobit dynamický text svázaný s proměnnou, je součástí DUMu č. 17.

Reakce na skok v časové ose

Studentům obvykle ukážeme společně, na kantorském počítači.

Vepíšeme-li do Actions našeho čtverce kód:

```
onClipEvent(enterFrame) {trace ("hop");}
```

bude se do okna Output pravidelně vypisovat slovo „hop“ při každém skoku na aktuální snímek. Není-li řečeno jinak, skáče Flash pouze po aktuálním snímku, stále dokola, rychlostí, jakou zadáme ve vlastnostech animace. Vlastnostmi animace začíná 3. DUM.