2 PRÁCE S UDÁLOSTMI A FUNKCEMI

Obsah lekce

V této lekci se naučíte následující:

- Používat fragmenty kódu pro tvorbu kódu jazyka ActionScript, který způsobí posun na časové ose na základě klepnutí na tlačítko.
- Přidávat kód do funkce vytvořené z fragmentu kódu.
- Psát posluchače událostí pro naslouchání událostem myši.
- Psát obsluhující funkce, které budou reagovat na události myši.
- Spojovat textové řetězce s hodnotami proměnných pro naplnění textového pole.
- Vytvářet a volat funkci, jež nastaví jazyk v textovém poli.
- Používat tlačítka ke změně hodnoty proměnné.

Tato lekce zabere přibližně dvě hodiny.

V předchozí lekci jsme vytvářeli zdrojový kód přímo ve snímcích na časové ose, a tento kód se spustil automaticky při přehrávání snímku, v němž se nacházel. Rovněž jsme začali pracovat s událostmi, když jsme přidali fragment kódu k souboru lekce 1. V této lekci se podrobněji seznámíte s událostmi v jazyce ActionScript. Pochopení modelu událostí je pravděpodobně nejdůležitější krok k ovládnutí základů jazyka a k tvorbě RIA aplikací.

Jazyk ActionScript 3.0 má spoustu vestavěných událostí a po spuštění události lze provést spoustu akcí. Při učení se jazyku ActionScript je nutné velkou část času věnovat dostupným událostem a zjišťování, jak reagovat na spuštění události. Až se blíže seznámíte s jazykem ActionScript, budete vytvářet své vlastní události.

Události a funkce jazyka ActionScript vytváří více interaktivních možností pro nás a naše uživatele.

Způsob práce s událostmi není nijak složitý. Napíšeme kód, kterým sdělujeme objektu, aby naslouchal události, a také napíšeme funkci, která se má zavolat v reakci na tuto událost. Na rozdíl od jazyků ActionScript 1 a 2 se syntaxe pro naslouchání a odpovídání na události shoduje v celém jazyce ActionScript 3. Začínající vývojáři však můžou mít problém naučit se tuto syntaxi. Dobré zprávy jsou, že v prostředí Flash CS5 můžeme vytvářet základní posluchače událostí pomocí panelu Fragmenty kódu (Code Snippets). Tento panel rovněž představuje skvělý způsob, jak se seznámit se syntaxí jazyka ActionScript.

V této lekci nejprve použijeme fragment kódu pro vytvoření posluchače události, pomocí něhož budeme provádět posun v panelu Časová osa (Flash Timeline), jakmile uživatel klepne na tlačítko. Postupně se přesuneme k psaní svých vlastních funkcí pro naslouchání událostem.

Práce s obsluhujícími funkcemi

Naslouchání a odpovídání na událost v jazyce ActionScript představuje proces se dvěma částmi. Prostřednictvím jedné části kódu – metody `addEventListener()`, zahájíme naslouchání události z určitého objektu. Jinou částí kódu – *obsluhující funkcí*, reagujeme na spuštěnou událost.

Jestliže máme kupříkladu ve scéně tlačítko, můžeme chtít provést tři věci:

- Zobrazit nabídku, když nad něj uživatel přesune ukazatel myši.
- Schovat nabídku, jakmile z něj uživatel odsune ukazatel myši.
- Přejít ke snímku časové osy, když na něj uživatel klepne.

V tomto příkladu používáme pouze jediné tlačítko, ale naslouchám třem různým událostem (událostem `ROLL_OVER`, `ROLL_OUT` a `CLICK`) a můžou nastat tři různé akce v závislosti na tom, která událost nastala.

Nejprve se zaměříme na první událost z našeho příkladu – kdyby mělo dané tlačítko název instance `tlacitko1`, sdělili bychom jazyku ActionScript, aby naslouchal události `ROLL_OVER`, takto:

```
tlacitko1.addEventListener(MouseEvent.ROLL_OVER, zobrazNabidku);
```

Pro události `ROLL_OUT` a `CLICK` budeme mít podobný řádek kódu.

Metodou `addEventListener()` říkáme objektu, aby začal naslouchat konkrétní události. Jakmile jednou zavoláme metodu `addEventListener()`, naslouchá, dokud ji neodstraníme. První argument uvnitř závorek volání metody `addEventListener()` označuje, jaké události chceme naslouchat. V tomto případě naslouchám události `ROLL_OVER` z kategorie `MouseEvent`. Povšimněte si, že názvy událostí obsahují výhradně velká písmena a mezi jednotlivými slovy jsou podtržítka. Konvenci používání velkých písmen pro názvy události si začínající vývojáři zapamatují snadno. Pomáhá také s odhalováním chyb při sestavování souborů a celkově vzato se vyplatí si ji zapamatovat.

Za názvem události a čárkou následuje druhý argument, a to funkce, jež se má spustit, když nastane událost `ROLL_OVER`. *Funkce* je jen blok zdrojového kódu, který vykonává jednu nebo více (obvykle souvisejících) úloh. *Obsluhující funkce* je funkce, která reaguje na událost.

Funkcím můžeme dávat téměř jakákoliv jména, přičemž se řídíme stejnými třemi pravidly jako pro pojmenování proměnných v lekci 1, „Používání úryvků kódu a navigace v panelu Časová osa". V tomto příkladu jsme funkci pojmenovali `zobrazNabidku()`. Je vhodné dávat funkcím jména, která popisují, co funkce dělá.

Připomenutí pravidel pojmenovávání v jazyce ActionScript

Zapamatujte si, že při pojmenovávání proměnných, funkcí, tříd a instancí byste se měli řídit následujícími třemi pravidly:

- Používáme pouze písmena, čísla a podtržítka ve svých názvech; vyhýbáme se jiným speciálním znakům.
- Nevkládáme číslo na začátek názvu.
- Nevkládáme do názvů mezery; raději místo nich použijeme podtržítka.

Základní syntaxe naší funkce vypadá takto:

```
function zobrazNabidku(e:MouseEvent):void {
  // Veškerý zdrojový kód jazyka ActionScript pro zobrazení nabídky bychom
  // měli umístit právě sem - mezi levou a pravou složenou závorku.
}
```

Poznámka: Jazyk ActionScript vždy rozlišuje velikost písmen. Na názvech funkcí a proměnných v této knize si můžete všimnout běžné konvence zápisu, kdy název začíná malým písmenem a každé následující slovo v názvu začíná velkým písmenem – stejně jako vidíte na názvu `zobrazNabidku()`. Přestože dodržovat tuto konvenci není nutné, jedná se o obvyklý programátorský postup, kterému se často říká „velbloudí" zápis (*camelCase)*. Tento způsob zápisu pomáhá snadněji určit, s jakým prvkem pracujete. Měli byste zvážit, zda tuto konvenci začleníte do své práce.

Když vytváříme funkci v jazyce ActionScript 3.0, vždy začneme napsáním slova `function` malými písmeny a za ním uvedeme název funkce, který jsme vybrali. Potom přidáme dvojici závorek, jež obsahuje to, co nazýváme *parametry*. S parametry budeme pracovat více v následujících lekcích. Zatím si zapamatujte, že obsluhující funkce obsahuje odkaz na událost, jež spustila danou funkci.

Po závorkách následuje dvojtečka a za ní informace o tom, jaký typ dat tato funkce vrací. V tomto případě slovo `void` označuje, že funkce nevrací žádná data. Více informací o funkcích se dozvíte v následujících lekcích.

Dále se nachází dvojice složených závorek, v nichž je uveden zdrojový kód, který se provede pokaždé, když událost spustí tuto funkci.

Pokud vám to stále není zcela jasné, nezoufejte. S troškou praxe to začíná dávat stále větší smysl a za malou chvíli vám to přijde naprosto přirozené. Výsledek určitě stojí za námahu. Jak už jsme si řekli, naučit se pracovat s posluchači událostí a obsluhujícími funkcemi je pravděpodobně nejdůležitější krok při osvojování jazyka ActionScript 3.0, a tato technika je konzistentní napříč celým

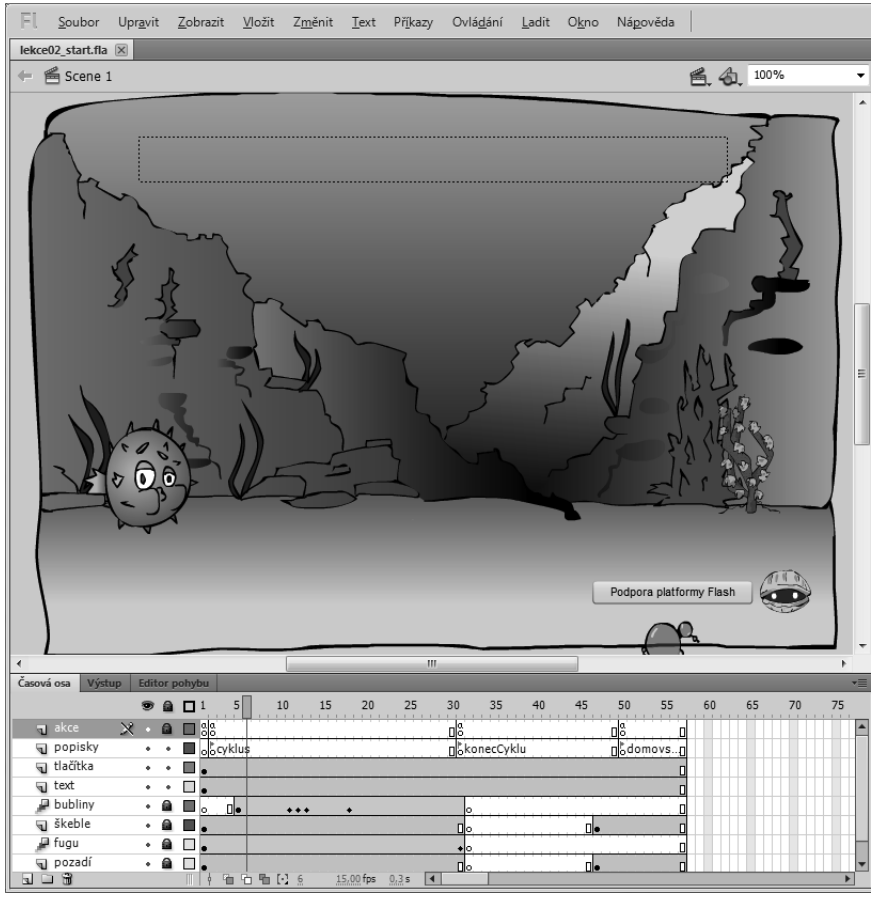
jazykem. To, co se naučíte v této lekci, bude vstupní branou k širokým interaktivním možnostem jazyka ActionScript 3.0.

Jak jsme viděli v lekci 1, pomocí fragmentů kódu je možné vytvářet funkce, jež reagují na události myši. Tuto lekci zahájíme tak, že použijeme fragment kódu k vytvoření funkce `eventListener()`, která provádí posun v panelu Časová osa (Flash Timeline), když uživatel klepne na tlačítko. Postupně pak začneme psát kód funkce `eventListener()` úplně sami.

Při své práci můžete stále používat fragmenty kódu jako výchozí body, nebo možná zjistíte, že je efektivnější, když si kód napíšete sami.

Používání fragmentů kódu k tvorbě navigace

Tuto lekci zahájíme se souborem z lekce 1. V prostředí Flash CS5 si otevřeme buď kompletní verzi tohoto souboru, nebo soubor lekce02_start.fla ze složky Lekce → Lekce02 → Start.



Vytváření instancí tlačítek k řízení navigace

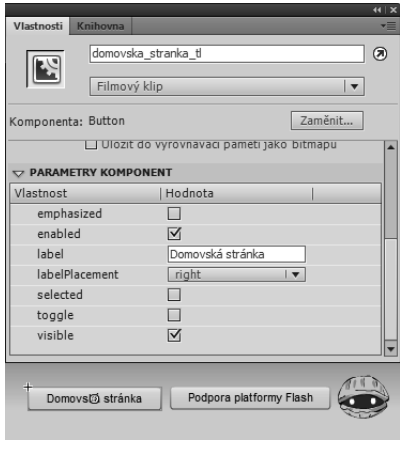
U většiny jednoduchých webových projektů Flash se většina interaktivity skládá z navigace spouštěné klepnutím na tlačítko. Schopnost psát kód jazyka ActionScript, ve kterém reagujeme na událost `CLICK` u tlačítka, je rovněž základem pro pochopení zbývajících částí jazyka ActionScript, protože všechny ostatní události fungují podobným způsobem.

Aby nám prostředí Flash CS5 usnadnilo začátky s touto důležitou funkčností, obsahuje sadu fragmentů kódu, jež automaticky napíšou kód pro navigaci v časové ose, kterou vyvolává klepnutí na tlačítko. Tyto fragmenty kódu budeme používat hned potom, co přidáme do našeho projektu nové tlačítko.

1. V panelu Časová osa (Flash Timeline) vybereme snímek 1 vrstvy tlačítka.
2. Pokud není vidět, otevřeme panel Knihovna (Library) (položkou Okno → Knihovna).
3. Přetáhneme instanci komponenty Button z panelu Knihovna (Library) vedle současného tlačítka Podpora platformy Flash v dolním pravém rohu scény.
4. Vybereme nové tlačítko ve scéně a v panelu Vlastnosti (Properties) (položka Okno → Vlastnosti) v části Parametry komponent (Component Parameters) najdeme vlastnost `label`.
5. Do pole vpravo od vlastnosti `label` zadáme text **Domovská stránka** a stiskneme klávesu Enter (operační systém Windows) nebo klávesu Return (operační systém Mac).

   Popisek tlačítka by se měl změnit na Domovská stránka. Toto tlačítko použijeme, abychom uživateli umožnili přechod ke snímku s domovskou stránkou.

   Nyní tomuto tlačítku přiřadíme název instance.
6. S vybraným novým tlačítkem přejdeme do panelu Vlastnosti (Properties), umístíme kurzor do pole Název instance a dáme tlačítku název instance `domovska_stranka_tl`.



Poznámka: Názvy instancí se řídí stejnými pravidly, která jsme si uvedli pro proměnné a funkce.

Význam názvů instancí

Je nezbytné dát názvy instancí všem tlačítkům, filmovým klipům a dalším objektům, které hodláme řídit s pomocí jazyka ActionScript. Nejčastější chybou, kterou dělají začínající programátoři jazyka ActionScript, je, že napíšou kód správně, ale zapomenou dát svým objektům názvy instancí. Kontrola názvů jmen je vždy dobrý začátek, když řešíme problémy s nefunkčním kódem.

Přidání fragmentu kódu pro navigaci

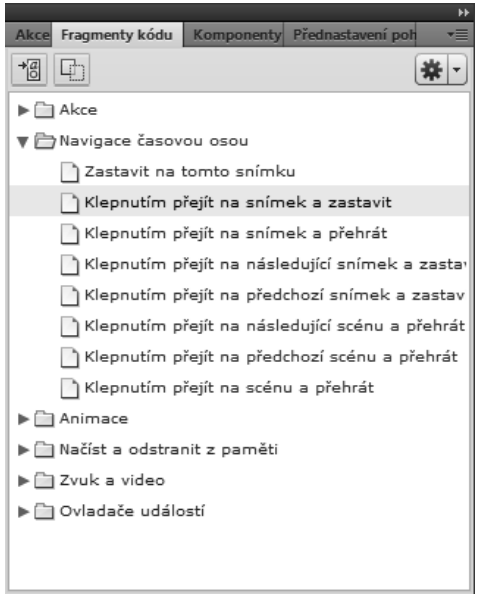
Účelem tlačítka Domovská stránka, jež jsme právě přidali, je umožnit uživateli přeskočit na snímek časové osy označený jako domovskaStranka, jestliže na něj uživatel klepne. Můžeme přidat nějaký zdrojový kód pomocí panelu Fragmenty kódu (Code Snippets).

1. Jestliže nejsou vidět, otevřeme panely Akce (Actions) (pomocí položky Okno → Akce) a Fragmenty kódu (Code Snippets) (položkou Okno → Fragmenty kódu (Code Snippets)).
2. Na časové ose vybereme snímek 2 ve vrstvě akce. Do tohoto snímku budeme vkládat fragment kódu.

O zdrojovém kódu v panelu Časová osa

V prvních dvou lekcích vkládáme svůj kód do více snímků panelu Časová osa. Jedná se o běžnou techniku u jednoduchých webových stránek založených na platformě Flash. U složitějších projektů se spousta programátorů vyhýbá vkládání kódu jazyka ActionScript do více snímků; místo toho se raději snaží napsat všechen kód do jediného snímku nebo použít externí soubory bez jakéhokoliv kódu v časové ose. Až se lépe seznámíte s jazykem ActionScript a tyto techniky vám nebudou činit potíže, můžete se sami rozhodnout, co nejlépe vyhovuje vám a vašemu projektu. Jak pracovat s externími soubory s kódem jazyka ActionScript, se dozvíte v lekci 4, „Vytváření kódu jazyka ActionScript v externích souborech." Interaktivní projekt, jenž nahraje obsah do jediného snímku, vytvoříte v lekci 5, „Nahrávání obsahu pomocí jazyka ActionScript a komponent."

3. Na scéně vybereme tlačítko Domovská stránka. Vzpomeňte si, že jste mu dali název instance `domovska_stranka_tl`.
4. V panelu Fragmenty kódu (Code Snippets) otevřeme složku Navigace časovou osou a poklepáme na fragment kódu s názvem Klepnutím přejít na snímek a zastavit.



Nyní bychom měli vidět následující kód v panelu Akce (Actions), a to pod kódem, který zde již byl:

```
/* Klepnutím přejít na snímek a zastavit
Po klepnutí na určenou instanci symbolu se přehrávací hlava přesune na
¬určený snímek na časové ose a film se zastaví.
Lze použít na hlavní časové ose nebo na časových osách filmových klipů.

Pokyny:
1. Nahraďte číslo 5 v uvedeném kódu číslem snímku, na který se má po
¬klepnutí na instanci symbolu přesunout přehrávací hlava.
*/

domovska_stranka_tl.addEventListener(MouseEvent.CLICK,
 ¬fl_ClickToGoToAndStopAtFrame);

function fl_ClickToGoToAndStopAtFrame(event:MouseEvent):void
{
  gotoAndStop(5);
}
```

Vytvořený fragment kódu přidal posluchače události k tlačítku Domovská stránka. Když uživatel klepne na toto tlačítko, automaticky se zavolá funkce, jejíž název je poněkud upovídaný – `fl_ClickToGoToAndStopAtFrame` (změna jejího názvu v kódu nijak neovlivní její chování a přesně to budeme dělat v následující části).

Jakmile se zavolá funkce, provede se veškerý zdrojový kód mezi jejími složenými závorkami. V tomto případě to znamená, že pokud uživatel klepne na tlačítko Domovská stránka, funkce posune časovou osu na snímek 5. Příkaz `gotoAndStop()` jsme již používali v lekci 1. Jediný rozdíl je, že tentokrát jej spouštíme v reakci na událost tlačítka. Snímek, k němuž tato funkce provádí posun, upravíme v následující části.

Úprava fragmentu kódu

Fragmenty kódu poskytují jednoduchý způsob, jak vytvořit správnou syntaxi jazyka ActionScript, ale jen zřídka nabízí přesně takovou funkčnost, jakou potřebujeme. Obyčejně si vybereme fragment kódu, jenž nejblíže odpovídá našim představám a upravíme jeho kód tak, aby vyhovoval našim potřebám. V právě vytvořeném fragmentu kódu provedeme několik úprav, aby se choval tak, jak chceme, a aby byl přehlednější.

Vzpomeňte si, že světle šedé znaky ve fragmentu kódu jsou jen popisné a nefunkční komentáře. Když si přečtete část komentáře s pokyny, zjistíte, že aby kód spustil navigaci k požadovanému snímku, musíte nahradit číslo `5` na řádku `gotoAndStop(5);` odkazem na snímek, na nějž má uživatel přejít po klepnutí na tlačítko. Jedním z možných řešení je změnit číslo `5` na jiné číslo. Ale lepším způsobem, jak se odkázat na snímek, je odkázat se na jeho popisek. Jestliže používáme popisky snímků místo čísel snímků ve svých skriptech, můžeme snadněji měnit obsah snímků časové osy, aniž bychom museli upravovat zdrojový kód. Popisky snímků předáme metodě `gotoAndStop()` tak, že nahradíme současné číslo `5` uvnitř závorek popiskem uzavřeným mezi uvozovky.

1 V panelu Akce (Actions) upravíme řádek `gotoAndStop(5)` takto:

```
gotoAndStop("domovskaStranka");
```

2 Uložíme si svou práci a otestujeme náš film (položkou Ovládání → Testovat film). Když při úvodní animaci klepneme na tlačítko Domovská stránka, časová osa by měla přeskočit rovnou na snímek domovskaStranka.

3 Zavřeme soubor lekce02_start.swf, abychom opustili testovací prostředí.

Nyní je zřejmé, jak lze jednoduše upravit fragment kódu, abychom dosáhli požadované navigace.

Tento kód bychom mohli v našem projektu ponechat v tomto stavu. Ale další úpravy usnadní práci s ním, jak náš projekt bude růst.

Jednou takovou změnou, kterou možná budeme chtít provést, je přejmenovat danou funkci. Nyní má tato funkce dlouhý a obecný název `fl_ClickToGoToAndStopAtFrame`. Praxí osvědčeným postupem je pojmenovávat funkce tak, aby názvy popisovaly jejich účel. Změňme tudíž název naší funkce na kratší a popisnější název `jdiDomu`. Tento název musíme změnit na dvou místech – ve volání metody `addEventListener()` a v samotné funkci.

4 V panelu Akce (Actions) vyhledáme řádek:

```
domovska_stranka_tl.addEventListener(MouseEvent.CLICK,
¬ fl_ClickToGoToAndStopAtFrame);
```

5 A změníme jej na řádek:

```
domovska_stranka_tl.addEventListener(MouseEvent.CLICK,
¬ jdiDomu);
```

6 Dále najdeme řádek:

```
function fl_ClickToGoToAndStopAtFrame(event:MouseEvent):void
```

7 A změníme jej takto:

```
function jdiDomu(event:MouseEvent):void
```

Změna názvu funkce tímto způsobem nemá žádný vliv na chování jejího kódu, ale je stručnější a lze jej snadněji pochopit.

Až několikrát použijete fragment kódu a pochopíte jeho komentář (šedá část kódu), možná budete chtít tento komentář smazat, což vůbec nevadí, protože to žádným způsobem neovlivní chování kódu.

8 Smažeme komentáře.

Zde je doposud finální podoba kódu pro snímek 2 bez komentářů:

```
info_txt.text = String(pocet);

domovska_stranka_tl.addEventListener(MouseEvent.CLICK,
¬ jdiDomu);

function jdiDomu(event:MouseEvent):void {
  gotoAndStop("domovskaStranka");
}
```

Jak se budete více sbližovat s jazykem ActionScript, budete chtít pravděpodobně tento kód ještě trochu upravit, aby byl lépe čitelný. Nyní ale pojďme dále.

Protože už jsme vytvořili navigaci pomocí tlačítka s použitím fragmentu kódu, zkusíme napsat podobný kód sami.

Vytváření posluchačů událostí

Přestože používat fragmenty kódu je pohodlné, abychom vytěžili z jazyka ActionScript 3.0 maximu, je absolutně nezbytné zcela pochopit jeho základní syntaxi. Tato schopnost přichází s časem, studiem a tréninkem. Až si osvojíte syntaxi jazyka ActionScript pro nějakou úlohu, stále vám mohou fragmenty kódu někdy významně ušetřit čas. Nyní je však čas, abychom napsali náš vlastní kód jazyka ActionScript sami. Začneme tvorbou dalšího posluchače události, přičemž zdrojový kód napíšeme od začátku.

Přidání tlačítka pro opětovné spuštění

Teď přidejme další funkčnost na domovskou stránku, abychom umožnili uživateli opětovně spustit úvodní animaci.

1 Přidáme klíčový snímek (klávesou F6) k vrstvě tlačítka na snímku domovskaStranka.

2 Klepneme kamkoliv do scény mimo naše dvě tlačítka, abychom zrušili jejich výběr a potom vybereme jen tlačítko Domovská stránka.

3 S vybraným tlačítkem Domovská stránka přejdeme do panelu Vlastnosti (Properties) (položka Okno → Vlastnosti) a změníme popisek tlačítka z Domovská stránka na **Spustit znovu**.

4 Když už jsme v panelu Vlastnosti (Properties), změníme také název instance tlačítka z `domovska_stranka_tl` na `restart_tl`.

5 S otevřeným panelem Akce (Actions) vybereme snímek domovské stránky ve vrstvě akce.

6 Přidáme následující zdrojový kód do panelu Akce (Actions) těsně pod současný kód:

```
restart_tl.addEventListener(MouseEvent.CLICK, jdiStart);
```

Dávejte pozor na přesnou velikost písmen a všimněte si, že výrazy `addEventListener` a `MouseEvent.CLICK` zmodrají, když je napíšete správně.

Barevné zvýrazňování kódu jazyka ActionScript při psaní vám pomáhá se ujistit, že píšete kód správně. Klíčová slova jazyka ActionScript 3.0 mají ve výchozím nastavení modrou barvu. Jestliže píšete nějaké slovo, jež je součástí jazyka ActionScript a ono se objeví jako černý text, měli byste zkontrolovat zápis a velikost písmen.

● **Poznámka:** Nezáleží na tom, kolik prázdných řádků vložíme mezi části našeho zdrojového kódu. Spousta programátorů ponechává prázdné místo mezi částmi kódu kvůli přehlednosti; jiní programátoři mají raději zhuštěný zápis a zahajují nové bloky kódu na každém řádku. Jak se začnete více seznamovat s jazykem ActionScript, určitě si najdete styl, který vám bude nejlépe vyhovovat.

Jakmile tento kód přidáme, dojde při klepnutí na tlačítko Spustit znovu k pokusu o zavolání funkce `jdiStart()`. Takže teď přidáme tuto funkci, aby tak mohl náš kód správně fungovat.

7 V panelu Akce (Actions) přidáme následující kód hned pod kód, jež jsme právě přidali:

```
function jdiStart(e:MouseEvent):void {
  pocet = 1;
  gotoAndPlay("cyklus");
}
```

Tato funkce bude reagovat na klepnutí na tlačítko Spustit znovu. Po zavolání této funkce se animace vrátí na začátek a proměnnou `pocet` nastavujeme zpět na hodnotu `1`. Vzpomeňte si, že proměnná `pocet` ukládá počet přehrání úvodní animace. Nastavením proměnné počet na hodnotu `1` tedy vracíme film do jeho původního nastavení.

8 Uložíme si svou práci a otestujeme náš film.

Jakmile se v testovacím prostředí dostaneme ke snímku domovské stránky, změní se popisek tlačítka Domovská stránka na Spustit znovu. Tlačítko Spustit znovu by mělo reagovat na klepnutí tak, že přehraje úvodní animaci od začátku a změní se zpět na tlačítko Domovská stránka. Všimněte si, že tlačítko Podpora platformy Flash funguje po celou dobu stejně. Protože jsme nezměnili jeho název instance, vždy se k němu váže posluchač události a funkce, jež jsme vytvořili na snímku 1.

Pokud všechno funguje jak má, nezbývá nic jiného, než vám pogratulovat. Jste na nejlepší cestě k ovládnutí jazyka ActionScript 3.0. Jestliže jste měli problémy s kódem, pečlivě porovnejte svůj kód s ukázkovým kódem. Pokud kód obsahuje chyby, měl by se objevit panel Výstup (Output) s popisy chyb a rovněž by měl ukázat, na kterých řádcích se vyskytly. Povšimněte si, která čísla řádků obsahují chyby, následně zkontrolujte zápis a barevné zvýraznění kódu jazyka ActionScript

na těchto řádcích. Zejména dávejte pozor na velikost písmen a ujistěte se, že názvy instancí vašich tlačítek odpovídají názvům uvedeným v posluchačích událostí.

Dynamická změna textového pole

Když nyní přehráváme úvodní animaci, textové pole zobrazuje číslo představující počet přehrání této animace. Číslo je sice přesné, ale informace předaná uživateli není příliš elegantní.

Proto uděláme tuto informaci užitečnější tak, že do textového pole přidáme několik dalších slov, abychom tak měli kompletní větu.

1 Se zobrazenými panely Akce (Actions) a Časová osa (Flash Timeline) vybereme snímek cyklus (snímek 2) ve vrstvě akce na časové ose.

2 V panelu Akce (Actions) změníme tento řádek kódu:

```
info_txt.text = String(pocet);
```

na následující řádek kódu:

```
info_txt.text = "Animace se přehrála " + String(pocet) + "x.";
```

Pomocí znaménka plus zřetězujeme (spojujeme) prostý text (v uvozovkách) s hodnotou proměnné `pocet` tak, abychom vytvořili větu.

3 Uložíme si práci a otestujeme náš film ještě jednou. Textové pole by mělo nyní obsahovat text: „Animace se přehrála 1x (2x, 3x atd.).“

Přidání tlačítek pro ovládání jazyka

Abyste si upevnili dosud získané znalosti, můžete zkusit přidat několik tlačítek do scény, s jejichž pomocí umožníte uživateli kontrolovat jazyk zobrazený v textovém poli. Začneme přidáním proměnné, jež uchovává uživatelem zvolený jazyk a nastavuje výchozí jazyk pro první snímek filmu.

1. Se zobrazenými panely Akce (Actions) a Časová osa vybereme snímek 1 ve vrstvě akce.
2. Přidáme následující zdrojový kód pod současný kód:

   ```
   var jazyk:String = "Čeština";
   ```

   Nyní přidáme kód, který bude ověřovat hodnotu proměnné `jazyk` předtím, než vloží text do textového pole.
3. S otevřenými panely Akce (Actions) a Časová osa vybereme snímek 2 ve vrstvě akce.
4. V panelu Akce (Actions) na snímku 2 vybereme tento řádek kódu:

   ```
   info_txt.text = "Animace se přehrála " + String(pocet) + "x.";
   ```

   a vyjmeme jej do schránky (klávesovou zkratkou Ctrl+X v operačním systému Windows nebo klávesovou zkratkou Command+X v operačním systému Mac).
5. Umístíme kurzor v panelu Akce pod poslední řádek zde uloženého zdrojového kódu.
6. Vytvoříme novou funkci, v níž budeme kontrolovat nastavený jazyk, a to tak, že přidáme následující kód do panelu Akce:

   ```
   function nastavJazyk():void {
   ```

```
  if (jazyk == "Čeština") {
  }
}
```

7 Na řádek před první pravou složenou závorku (}) vložíme kód, který jsme vyjmuli, takže naše funkce bude vypadat takto:

```
function nastavJazyk():void {
  if (jazyk == "Čeština") {
    info_txt.text = "Animace se přehrála " + String(pocet) +
    ¬ "x.";
  }
}
```

● **Poznámka:** Pomocí podmíněného příkazu ve funkci `nastavJazyk()` kontrolujeme, jestli je jazyk nastaven na češtinu. Toto porovnávání provádíme pomocí dvou znamének rovnosti (==).

V jazyce ActionScript 3.0 ověřujeme, že jedna hodnota odpovídá jiné hodnotě pomocí dvou znamének rovnosti. V tomto případě kontrolujeme, zda se hodnota proměnné `jazyk` rovná hodnotě `"Čeština"`.

Je velmi důležité zapamatovat si, že hodnoty porovnáváme pomocí dvou znamének rovnosti, protože jedno znaménko rovnosti (=) by znamenalo, že by se jedna hodnota nastavila tak, aby se rovnala druhé. Jinými slovy, v tomto příkladu by jedno znaménko rovnosti nastavilo proměnné `jazyk` hodnotu `"Čeština"` místo toho, aby ověřilo, že proměnná `jazyk` má hodnotu `"Čeština"`.

Tato funkce při svém zavolání zkontroluje, že proměnná `jazyk` má hodnotu `"Čeština"` (což je výchozí jazyk díky kódu, jenž jsme přidali v kroku 2). Jestliže je jazykem čeština, textové pole zobrazí naši zprávu.

Brzy přidáme tlačítka, s nimiž si uživatel bude moci vybrat češtinu, angličtinu nebo němčinu, proto vložíme do daného podmíněného příkazu další podmínky.

8 Rozšíříme funkci `nastavJazyk()` takto:

```
function nastavJazyk():void {
  if (jazyk == "Čeština") {
    info_txt.text = "Animace se přehrála " + String(pocet) +
    ¬ "x.";
  } else if (jazyk == "Angličtina") {
    info_txt.text = "The animation has played " + String(pocet)
    ¬ + "x.";
  } else if (jazyk == "Němčina") {
    info_txt.text = "Die Animation wurde " + String(pocet) +
    ¬ "x abgespielt.";
  }
}
```

Na rozdíl od funkcí, které jsme vytvořili dříve, není funkce `nastavJazyk()` obsluhující funkcí – tj. není určena k tomu, aby reagovala na určitý typ události. Je tomu tak kvůli tomu, že se tato funkce musí spustit při spuštění aplikace a pokaždé, když uživatel změní vybraný jazyk.

● **Poznámka:** V projektech Flash s velkým množstvím obsahu pro různé jazyky by bylo vhodnější uložit přeložené texty do externího umístění (například do souboru XML) a nahrát je do prostředí Flash za běhu. S externími soubory ve formátu XML se naučíte pracovat v pozdějších lekcích.

Tuto funkci zavoláme tak, že uvedeme její název a za něj napíšeme dvojici závorek. Kdyby funkce měla nějaké parametry, zapsali bychom jejich hodnoty do závorek. Tato funkce však žádné parametry nemá.

9 V panelu Akce (Actions) vybereme řádek za funkcí `nastavJazyk()`.

10 Zavoláme funkci `nastavJazyk()`, aby nastavila správně text na začátku animačního cyklu, a to napsáním níže uvedeného zdrojového kódu:

```
nastavJazyk();
```

Nakonec přidáme tlačítka, s nimiž bude moci uživatel měnit jazyk.

11 Vybereme snímek 1 vrstvy tlačítka v panelu Časová osa (Flash Timeline).

12 V panelu Knihovna (Library) (položka Okno → Knihovna) uvidíte tři tlačítka s názvy Tlačítko Čeština, Tlačítko Angličtina a Tlačítko Němčina. Přetáhneme instanci každého z nich do levého horního rohu scény. Jedná se o běžná tlačítka s přidaným textem.

13 V panelu Vlastnosti (Properties) pojmenujeme instance nových tlačítek `cestina_tl`, `anglictina_tl` a `nemcina_tl`.

14 Vrátíme se ke snímku 2 vrstvy akce a přidáme posluchač události pro všechna tato tlačítka napsáním následujícího zdrojového kódu pod poslední uvedený řádek:

```
cestina_tl.addEventListener(MouseEvent.CLICK, nastavCestinu);
anglictina_tl.addEventListener(MouseEvent.CLICK,
¬ nastavAnglictinu);
nemcina_tl.addEventListener(MouseEvent.CLICK,
¬ nastavNemcinu);
```

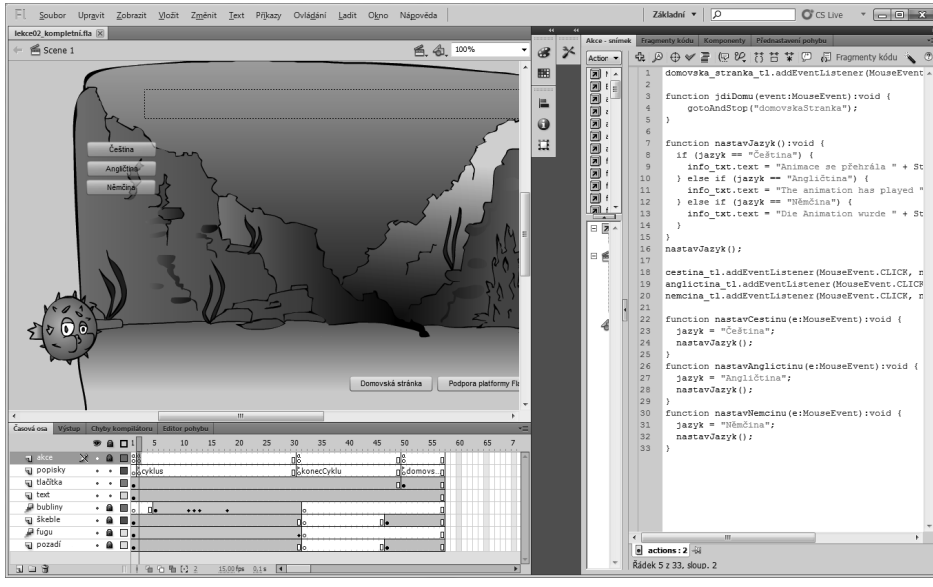
Když uživatel klepne na některé z těchto tlačítek, musíme provést dvě věci:

- Nastavíme proměnnou `jazyk` na vybraný jazyk.
- Zavoláme funkci `nastavJazyk()`, jež změní obsah textového pole.

Vzpomeňte si, že podmíněný příkaz ve funkci `nastavJazyk()` používá hodnotu proměnné `jazyk`, aby rozhodl o tom, co má vepsat do textového pole.

15 Na řádky pod právě vytvořené posluchače událostí přidáme níže uvedený zdrojový kód:

```
function nastavCestinu(e:MouseEvent):void {
  jazyk = "Čeština";
  nastavJazyk();
}
function nastavAnglictinu(e:MouseEvent):void {
  jazyk = "Angličtina";
  nastavJazyk();
}
function nastavNemcinu(e:MouseEvent):void {
  jazyk = "Němčina";
  nastavJazyk();
}
```



16 Uložíme si svou práci a otestujeme náš film.

Textové pole se nejdříve zobrazí v češtině. Během přehrávání úvodní animace bychom měli mít možnost přepínat obsah textového pole mezi jazyky čeština, angličtina a němčina. Pokud klepneme na tlačítko Spustit znovu, aktuálně vybraný jazyk zůstane zachovaný, dokud jej nezměníme (klepnutím na jiné tlačítko).

Několik návrhů pro vyzkoušení

Pokud jste dočetli až sem, nezbývá, než vám poblahopřát. Jste opravdu dobří studenti a pravděpodobně vás překvapuje, čeho jste schopni dosáhnout pouze s technikami uvedenými v prvních dvou lekcích.

Abyste se lépe seznámili s technikami z těchto kapitol, můžete zkusit přidat několik funkcí do souboru lekce02_start.fla. Zde je několik příkladů:

- Přidejte další jazyky. K tomu budete muset přidat nová tlačítka, posluchače událostí a funkce do stávajícího kódu jazyka ActionScript. Použijte jakékoliv jazyky, které znáte, přičemž k překladu textů můžete použít webové stránky na adrese *http://www.freetranslation.com*, nebo si udělejte vlastní překlad.
- Přeložte text na snímku domovské stránky. Dosud jste přeložili jen obsah textového pole v průběhu úvodní animace, ale pro snímek domovské stránky můžete napsat podobnou funkci, jež přeloží text na základě uživatelem zvoleného jazyka.
- Použijte kód jazyka ActionScript podobný tomu, který jste přidali k tlačítku Podpora platformy Flash; přidejte další tlačítka s odkazy na jiné adresy URL.
- Použijte podobný kód jazyka ActionScript jako u tlačítka Domovská stránka – přidejte tlačítka, která přejdou a zastaví na určitých snímcích nebo přejdou a přehrají určité snímky animace.

Opakování

Otázky

1. Popište, k čemu slouží v jazyce ActionScript 3.0 metoda `AddEventListener()`.
2. Jakým způsobem lze zapsat událost klepnutí myší v metodě `addEventListener()`?
3. Jaký znak lze v jazyce ActionScript 3.0 použít pro spojování (nebo zřetězování) textových řetězců a názvů proměnných?
4. Jaký zápis použijete pro kontrolu, zda se jedna hodnota rovná druhé? Jakým zápisem nastavíte proměnné danou hodnotu?

Odpovědi

1. Metoda `addEventListener()` slouží pro naslouchání určité události na konkrétním objektu a reaguje na událost zavoláním obsluhující funkce.
2. V metodě `addEventListener()` lze událost klepnutí myší zapsat jako `MouseEvent.CLICK`, jak je patrné zde:

   ```
   Tlacitko1.addEventListener(MouseEvent.CLICK, udelejNeco);
   ```

3. Pomocí znaménka plus (+) lze zřetězit text s vyhodnoceným výrazem. To běžně slouží k nastavení vlastnosti `text` dynamického textového pole. Zde je příklad:

   ```
   nejakeTextovePole.text = "Ahoj " + jmenoUzivatele;
   ```

4. Prostřednictvím dvou znamének rovnosti můžete porovnat dvě hodnoty, abyste zjistili, zda jsou stejné, jak vidíte zde:

   ```
   if (heslo == 3456789) {
     bezpecneVstup();
   }
   ```

   Jedno znaménko rovnosti slouží pro nastavení hodnoty proměnné:

   ```
   var prvniPrezidentCR:String = "Havel";
   ```