



Elemente Multimedia pentru Programarea Web

Ciprian Dobre
ciprian.dobre@cs.pub.ro



Obiective

- În cadrul cursului prezentăm noțiuni introductive legate de programarea elementelor multimedia în paginile Web.

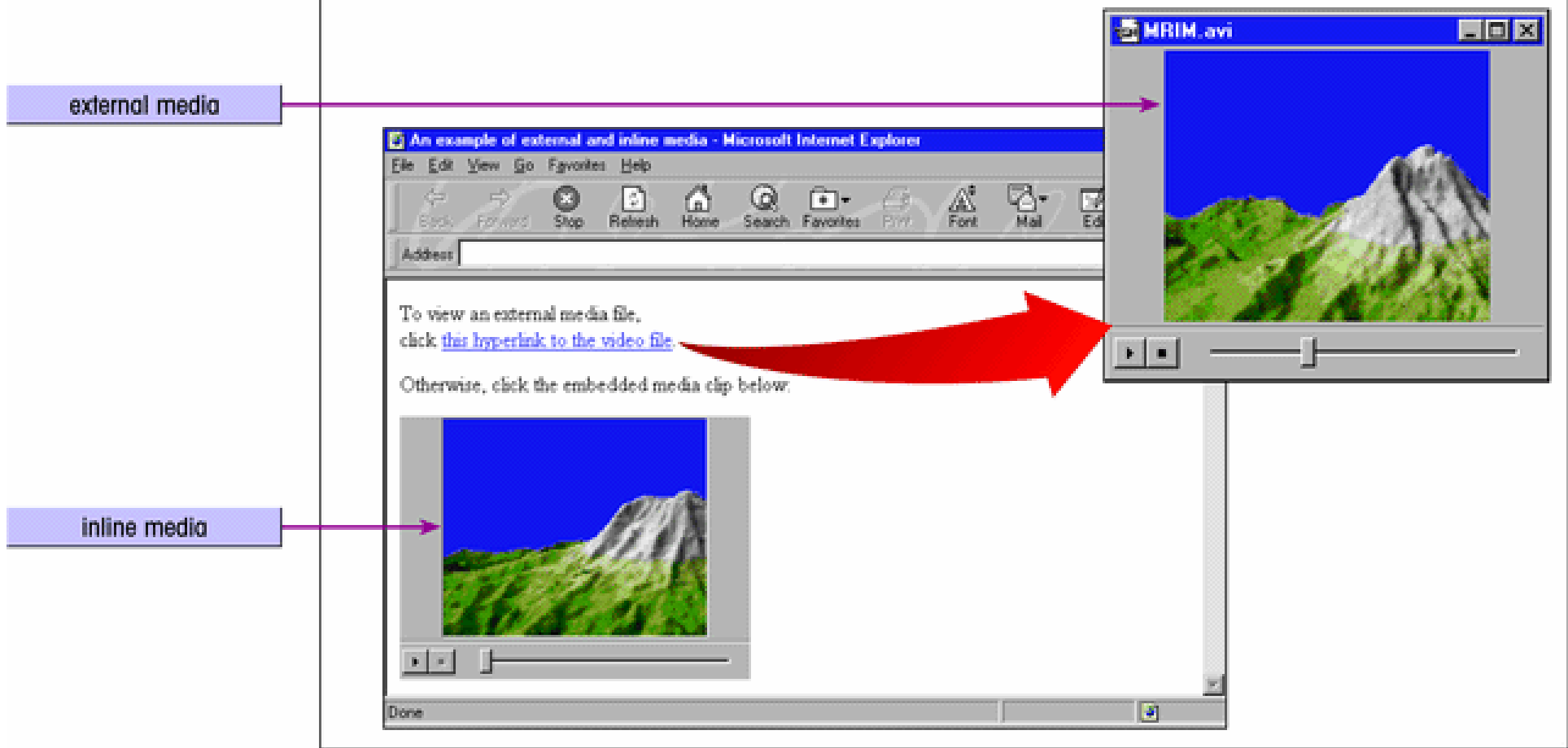


Multimedia

- Popular și folositor
- Bandwidth
- Elementele Multimedia pot fi adăugate în două feluri
 - External media: fișierul sunet sau video este accesat ca un hyperlink
 - Avantaj: Utilizatorii accesează clip-ul multimedia doar dacă doresc
 - Inline media: clipurile sunt plasate în interiorul paginii Web ca obiecte embedded
 - Avantaj: Material suplimentar în cadrul paginii
 - Dezavantaj: Utilizatorul este forțat să aștepte încărcarea clipului

Figure 10-1

INLINE AND EXTERNAL MEDIA





Fișiere Sunet

- Două componente ale undei sonore:
 - Amplitudinea: înălțimea undei sonore
 - Cu cât amplitudinea e mai mare cu atât volumul sunetului e mai tare
 - Frecvența: viteza cu care se mișcă unda sonoră
 - Frecvențele mai mari au înălțimea (pitch) mai mare
- O undă sonoră este o funcție continuă
 - Pentru convertirea ei la o formă ce poate fi stocată într-un fișier de sunet calculatorul trebuie să înregistreze măsurători ale sunetului la momente discrete de timp
 - Fiecare măsurătoare reprezintă un sample
 - Rata de măsurare se specifică în *samples per second*
 - Cele mai comune sunt 11 KHz, 22 KHz și 44 KHz
 - Calitate vs. dimensiunea fișierului



Fișiere Sunet

- Un al doilea factor în convergerea unui sunet într-un fișier digital este rezoluția de sampling
 - Indică precizia măsurătorii sunetului în fiecare sample
 - Două valori ale rezoluției: 8-bit și 16-bit
 - Rezoluția 16-bit crează un fișier mai mare având mai multe detalii ale unde de sunet (calitatea mai mare)
- Alegerea finală constă în determinarea numărului de canale ce vor fi folosite
 - În mod normal stereo sau monaural, dar pot fi adăugate canale suplimentare



Fișiere Sunet

Sample Rate and Resolution	Sound Quality
8 KHz, 8-bit, mono	Telephone
22KHz, 16-bit, stereo	Radio
44 KHz, 16-bit, stereo	CD



Formate de Fișiere Sunet

- Există câteva formate de fișiere ce se folosesc pe Web și care furnizează diverse nivele de calitate și compresie a sunetului
 - AIFF/AIFC: pentru folosirea pe Apple
 - AU: (mu-law) format; mai vechi, UNIX
 - MIDI: limitat la sintetizatoare și fișiere de muzică
 - MPEG: folosit în principal pentru video; MP3, mai mic
 - RealAudio: pentru redare real-time; bandwidth variabil; tehnologie de streaming media
 - SND: Macintosh; nu e suportat pe scară largă pe Web
 - WAV: dezvoltat pentru Windows; cel mai comun format pe Web



Arhive Sunet pe Web

Title	URL	Description
Broadcast.com	www.broadcast.com	A collection of live and archived recordings that use streaming media
Historic sound clips	www.webcorp.com/sounds	Sound clips from historical figures and events in history
Index of sounds	www.sunsite.sut.ac.jp/multimed/sounds	An archive of sound clips broken down into categories
MP3.com	www.mp3.com	News and information about MP3s; links to archives
MSU voice library	www.lib.msu.edu/vincent	Selections from the G. Robert Vincent Voice library
Sound America	www.soundamerica.com	An archive of almost 30,000 sound clips broken down into categories



Legătura cu un Fișier Sunet

- Folosirea unui hyperlink
 - Când se declanșează browserul caută aplicația ajutorătoare pentru redare instalată pe calculator
 - Aceasta va interpreta și prezenta respectivul fișier

`Wild Mountain Thyme (211k – WAV format)`



Încapsularea unui fișier sunet

- Un fișier încapsulat (embedded) poate fi orice clip media, fișier, program sau alt obiect ce poate fi rulat sau vizualizat din pagina Web
- Browser-ul trebuie să îl suporte și trebuie să aibă acces la software (plug-in sau add-ons)
- Browser-ul încarcă plug-in-ul plus alte controale necesare pentru manipularea obiectului
- Plug-in-urile necesită ca utilizatorul să descarce și instaleze software suplimentar înainte de a putea vizualiza pagina Web
 - Utilizatorii ar putea prefera să nu vadă respectiva pagină



Folosirea **<EMBED>**

- Pentru încapsularea unui clip de sunet se folosește tag-ul `<embed>`

`<embed src=URL width=value height=value
autostart="startvalue">`

- *height* și *width* definesc dimensiunea obiectului și controalele obiectului
- *autostart* este folosit pentru a determina dacă sau nu browser-ul începe clip-ul încapsulat automat odată cu încărcarea paginii
 - True: pornește automat
 - False: permite utilizatorului începerea manuală
 - Comportamentul implicit variază de la browser la browser



Folosirea lui <EMBED>

<p>Listen to the sounds of <i>Adams & Davis</i></p>

<blockquote>

<embed src="AF2000.wav" width=145 height=40
autostart="false">

</blockquote>



Utilizarea **<BGSOUND>**

- Introdus în IE 3.0
- Folosit pentru sunete de fundal
 - **<BGSOUND src=URL loop=value>**
 - *loop* definește de câte ori clipul este rulat (1, 2, 3, ...) sau INFINITE
- Atenție: utilizatorul nu are controale

<bgsound src="AF2000.wav" loop=1>



Fișiere Video

- Afișarea de video a devenit deosebit de populară pe Web
 - Grad mai mare de interactivitate și furnizarea mai rapidă de mai multă informație
- Fișierele video pot fi foarte mari și poate fi dificil de lucrat cu ele
 - 30 secunde pot duce la fișiere de 10 MB
- Fișierele video pot fi create pornind de la o placă de captură video sau un software de animație
 - Proces time-consuming



Frame Rates și Codecs

- Un fișier video este compus din frame-uri sau imagini individuale
 - Cadrele rulate în secvență dau iluzia de mișcare
- Rata de rulare reprezintă numărul de cadre prezentate în fiecare unitate de timp (fps)
- Rate mari furnizează o redare mai clară dar fișierele rezultate pot fi mai mari
 - VCR este 30 fps
- Dimensiunea poate fi controlată de compresia cadrelor
 - Compresia și decompresia cadrelor video se numește *codec*
 - Se poate reduce dimensiunea fișierelor video prin reducerea dimensiunii cadrelor video (160 x 120 pixeli este considerat standard pe Web)
 - Color vs. grayscale și sunetul pot influența de asemenea dimensiunea fișierului rezultat



Formate de Fișiere Video

Format	Description
ASF	Advanced Streaming Format developed by MS to replace AVI; employs streaming technology
AVI	Audi Video Interactive is a standard video format for Windows; resolution no larger than 320 x 240 pixels and no faster than 30 fps
MPEG	Moving Pictures Group allows for high compression; MPEG-1 352 x 240 pixels at 30 fps, MPEG-2 1280 x 720 at 60 fps
QuickTime	By Apple for the Macintosh; can play on Windows with special software
RealVideo	By RealNetworks; uses streaming technology to provide live video over low and high bandwidth connections; quality is usually poorer



Formate de fișiere video

- Pe care ar trebui să îl folosiți în pagina Web?
 - Depinde...
- Hiperlyng la un fișier video
 - `Summit Flyby (187k – AVI format)
`
 - `Summit Flyby (215k – QuickTime format)
`
- Încapsularea unui fișier video
 - `<embed src="MRIM..avi" width=200 height=200 autostart="false">`
- **<NOEMBED>** este folosit pentru a suporta browsere mai vechi ce nu suportă obiecte embedded
 - `<embed ...>`
 - `<noembed>`
 - html tags
 - `</noembed>`



Java Applets

- Java a fost dezvoltată pentru a permite utilizatorilor rularea de programe din interiorul browserelor Web
 - Fiecare astfel de program este numit un *applet*
- Spre deosebire de JavaScript, un applet Java nu este inserat în interiorul fișierului HTML
 - Reprezintă un fișier extern descărcat și executat de către browser
- Applet-ul apare de obicei ca un obiect încapsulat în pagina Web într-o fereastră numită *applet window*
 - Se pot specifica dimensiunea și poziția acestei ferestre
 - Unele applet-uri pot apărea în afara browser-ului ca o fereastră separată redimensionabilă – însă mai corect de folosit în astfel de cazuri ar fi tehnologia JavaWS



Arhive de Applet-uri Java

Title	URL
Applets from Sun	Java.sun.com/applets/
Gamelan	www.gamelan.com
Java Boutique	www.javaboutique.com
Java Rating Service	www.jars.com
Yahoo's list of Java applets	www.yahoo.com/Computers_and_Internet/Programming_Languages/Java/Applets



Tag-urile **<APPLET>** și **<PARAM>**

- Tag-ul `<applet>` se folosește pentru inserarea unui applet în pagina Web
 - Tag-ul identifică fișierul `.class` care va fi folosit și permite specificarea parametrilor necesari pentru execuția fișierului `.class`

```
<applet code=file.class width=value  
height=value>
```

```
<param name=text value=value>
```

```
<param name=text value=value>
```

```
</applet>
```



Proprietăți ale APPLET

Property	Description
ALT=text	A text string that is displayed in place of the applet before the applet has loaded
CODEBASE=URL	The location of the .class file if different from the Web page
CODE=filename.class	The filename of the .class file
HEIGHT=value	Height in pixels of the embedded applet
HSPACE=value	Horizontal space between the embedded applet and the surrounding text, in pixels
NAME=text	The name of the applet
VSPACE=value	The vertical space between the embedded applet and the surrounding text, in pixels
WIDTH=value	The width of the embedded applet in pixels



Exemplu de APPLET

```
<applet code="CreditRoll.class" width=400 height=60>
```

```
<param name=BGCOLOR value="FFFFFF">
```

```
<param name=TEXTCOLOR value="663366">
```

```
<param name=FADEZONE value=20>
```

```
<param name=VSPACE value=3>
```

```
<param name=SPEED value=100>
```

```
</applet>
```



Flash

- Flash este una dintre cele mai importante platforme pentru dezvoltarea de jocuri online
- Flash MX 2004 ActionScript are toată puterea oricărui limbaj de programare (non-web) profesional

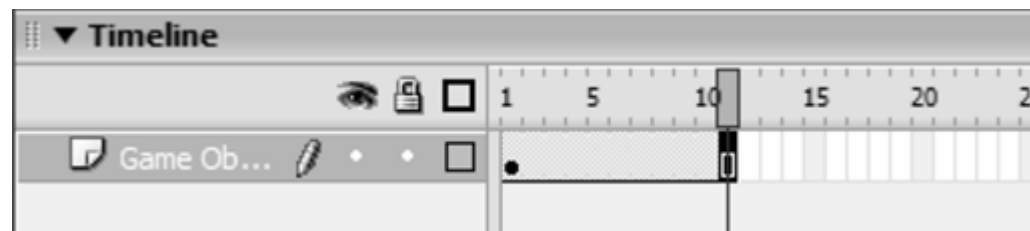


Avantaje Flash

- Audiență mare
 - 98% dintre utilizatorii web au player-ul Flash instalat
- Dezvoltare rapidă
- Deployment ușor
 - Flash crează automat pagini web
- ...

Cum funcționează Flash

- Timeline Based
 - Flash folosește un timeline de cadre ce face ca jocul să ruleze la o rată fixă de cadre





Cum funcționează Flash

- Timeline Based
 - Codul este plasat în cadre
 - Proiectanții și artiștii mai probabil vor folosi linia timpului pentru a desena și anima caracterele, grafica de mișcare, etc
 - Programatorii, pe de altă parte, au tendința de a plasa tot codul ActionScript într-un singur cadru și a ține propriul timeline lung de un singur cadru

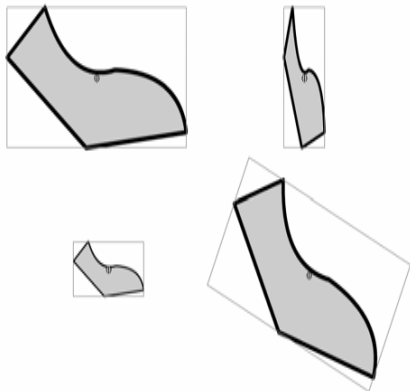
Cum funcționează Flash

- Vector Engine
 - Motorul grafic de bază din spatele Flash este un motor vectorial
 - Toată grafica și formele sunt definite în termenii unor forme matematice precum linii, curbe spline, cercuri și informație de stroke / fill
 - Avantajul unui motor vectorial este acela că jocul poate fi scalat la oricare dimensiune, iar grafica va continua să arate clară



Cum funcționează Flash

- Vector Engine
 - Formele pot fi umplute cu culori solidem, gradienti radială, liniară și chiar bitmap-uri
 - Cu cât vectorul este mai complex cu atât mai greu va fi randat de Flash și va duce la o degradare a performanțelor de rulare
 - Odată ce o formă este definită poate fi convertită într-un *movie clip*





Cum funcționează Flash

- Vector Engine
 - Clipurile video (Movie Clips) sunt elementele de bază ale jocurilor/interacțiunilor
 - Clipurile pot fi mutate, rotite, scalate și făcute transparente folosind cod ActionScript



Cum funcționează Flash

- Audio Engine
 - Putem crea efecte sonore, muzică sau adăuga voce în cadrul aplicațiilor și apoi importa aceste sunete în Flash
 - Putem crea o instanță a obiectului Sound în ActionScript și declanșa un sunet de câte ori avem nevoie de el
 - Sunetele pot fi comprimate ca raw, ADPCM sau MP3



Mediul de Scripting

- ActionScript
 - Toată codarea și scripting-ul sunt făcute în Flash într-un limbaj cunoscut ca ActionScript
 - Acolade pentru blocuri de Instrucțiuni { }
 - Punct și virgulă la sfârșitul liniilor ;
 - Condiții logice *if*
 - `if (condition) doSomething();`
 - Bucle



Mediul de Scripting

- ActionScript
 - Bucle. Buclele sunt formate folosind instrucțiuni standard for, do sau while.
 - `for (j = 0; j < 100; j++)`
 - {
 - do something 100 times;*
 - }



Mediul de Scripting

- ActionScript
 - Tipuri de Variabile
 - Standarde
 - Number
 - String
 - Array
 - Boolean



Mediul de Scripting

- ActionScript
 - Tipuri de variabile
 - Specifice Flash
 - Object
 - MovieClip
 - Color
 - Date
 - Camera
 - Microphone
 - Sound
 - TextField



Mediul de Scripting

- ActionScript
 - Clase
 - În Flash MX 2004, Macromedia a introdus ActionScript 2.0
 - ActionScript 2.0 a introdus abilitatea de a crea clase în adevăratul sens cu toate elementele standard de specifice altor limbaje OOP
 - Create în propriile fișiere numite fișiere clasă, având formatul “*classname.as*”



Mediul de Scripting

- ActionScript
 - API-ul de desenare
 - Flash folosește o serie de comenzi ActionScript disponibile la runtime numite Draw API
 - Draw API permite desenarea de forme și grafică la runtime folosind câteva rutine
 - moveTo, lineTo, lineStyle, beginFill, endFill



Integrarea Graficii și Codului

- Main Timeline
 - Aici avem câteva clipuri video și putem plasa cod ActionScript în cadrul unu al principalului timeline al oricărui clip

```
beachball._alpha = 10;  
beachball._rotation = 90;  
beachball._x = 214;  
beachball._y = 12;  
beachball._xscale = 50;  
beachball._yscale = 50;  
beachball.onPress = function()  
{  
    this._y++;  
}
```



Integrarea Graficii și Codului

- Movie Clip timeline
 - Este posibil și să plasăm cod pe primul cadru al oricărui (principal) clip video

```
this._alpha = 10;  
this._rotation = 90;  
this._x = 214;  
this._y = 12;  
this._xscale = 50;  
this._yscale = 50;  
this.onPress = function()  
{  
    this._y++;  
}
```



Action Script

Terminologia ActionScript

- **Acțiuni (Actions)** sunt instrucțiuni care comunică filmului să execute ceva în timp ce rulează. De exemplu, **gotoAndStop** trimite capul de redare la o anumită etichetă sau cadru.
- **Boolean** este o valoare adevărată (true) sau falsă.
- **Clasele** sunt tipuri de date (data types) create pentru a defini un tip nou de obiect. Pentru a defini o clasă se crează o funcție constructor.
- **Constantele** sunt elemente care nu se modifică. De exemplu, constanta `Key.TAB` are întotdeauna aceeași semnificație: indică tasta 'Tab'. Constantele sunt utile pentru compararea valorilor.



Action Script

- **Constructorii (Constructors)** sunt funcții utilizate pentru a defini proprietățile și metodele unei clase. De exemplu, următorul cod creează o nouă clasă 'Circle' prin crearea unei funcții constructor denumită Circle:
- `function Circle(x, y, radius){`
 `this.x = x;`
 `this.y = y;`
 `this.radius = radius;}`
- **Tipuri de date (Data types)** sunt seturi de valori și operații ce pot fi aplicate acestora. the operations that can be performed on them. Tipurile de date ale ActionScript sunt: **string, number, boolean, object, movieclip, function, null, și undefined.**



Action Script

- **Evenimentele (Events)** sunt acțiuni care au loc în timpul execuției unui film. De exemplu, la încărcarea (*load*) filmului sunt generate, capul de redare intră într-un cadru, utilizatorul execută clic pe un buton sau pe un 'MovieClip' sau utilizatorul apasă un șir de taste.
- **Event handlers (gestionarea de evenimente)** reprezintă acțiuni speciale care gestionează evenimente, cum ar fi *mouseDown* sau *load*. Există două feluri de gestionări de evenimente: gestionari de acțiuni și de metode. Există numai doi gestionari acțiuni: *on* și *onClipEvent*. În Actions toolbox, fiecare obiect ActionScript care are gestionar de metode, conține o subcategorie, denumită **Events**.
- **Expresiile** reprezintă orice combinație legală de simboluri ActionScript care reprezintă o valoare. O expresie constă din operatori și operanzi. De exemplu, în expresia $x + 2$, x și 2 sunt operandă iar $+$ este un operator.



Action Script

- **Funcțiile** sunt blocuri de cod reutilizabil, care poate transfera parametri și poate returna o valoare. De exemplu, funcția `getProperty` transferă numele unei proprietăți și numele instanței unui 'movie clip' și returnează valoarea respectivei proprietăți. Funcția `getVersion` returnează versiune Flash Player-ului ce execută 'Movie Clip'-ul în mod curent.
- **Identificatorii** sunt nume utilizate pentru a indica o variabilă, o proprietate, obiect, funcție sau metodă. Primul caracter trebuie să fie o literă, underscore (`_`) sau semnul dolarului (`$`). Fiecare caracter următor trebuie să fie o literă, număr, underscore sau `$`. De exemplu, `firstName` este numele unei variabile.
- **Instanțele** sunt obiecte aparținând unei anumite clase. Fiecare instanță a unei clase conține toate proprietățile și metodele respectivei clase. Toate 'movie clip'-urile sunt instanțe cu proprietăți (de exemplu, `_alpha` și `_visible`) și metode (de exemplu, `gotoAndPlay` și `getURL`) a clasei `MovieClip`.



Action Script

- **Numele instanțelor** sunt denumiri unice care permit apelarea clip-ului sau butonului țintă în cod. De exemplu, un simbol master în bibliotecă poate avea numele counter, iar două instanțe ale acestui simbol pot avea numele scorePlayer1 și scorePlayer2. Scriptul următor setează o variabilă denumită score în fiecare instanță a movie clip-ului prin utilizarea numelor instanțelor:
- `_root.scorePlayer1.score += 1;_root.scorePlayer2.score -= 1;`
- **Cuvinte cheie (Keywords)** sunt cuvinte rezervate cu semnificație specială. De exemplu, var este un cuvânt cheie utilizat pentru a declara o variabilă locală. Un cuvânt cheie nu poate fi utilizat ca identificator. De exemplu, var nu este un nume legal de variabilă.



Action Script

- **Metodele** sunt funcții alocate unui obiect. După alocarea unei funcții obiectului, acesta poate fi apelată ca metodă a respectivului obiect. De exemplu, în scriptul următor, `clear` devine metodă a obiectului `controller`:
- ```
function reset(){ this.x_pos = 0; this.x_pos = 0;}controller.clear = reset;controller.clear();
```
- **Obiectele** sunt colecții de proprietăți și metode; fiecare obiect are un nume și este o instanță a unei anumite clase. Obiectele 'Built-in' sunt predefinite în limbajul ActionScript. De exemplu, obiectul 'built-in' `Date` furnizează informația de la ceasul sistemului.
- **Operatori** sunt termeni care definesc operațiile de calcul a unor valori noi din una sau mai multe valori. De exemplu, operatorul de adunare (+) adună două sau mai multe valori pentru a obține o nouă valoare. Valorile manipulate de către operatori se numesc operanzi.



# Action Script

- **Parameterii** (denumiți și argumente) sunt placeholders care permit transmiterea valorilor către funcții. De exemplu, următoarea funcție welcome utilizează două valori pe care le primește în parametrii firstName și hobby:
- ```
function welcome(firstName, hobby) {  
    welcomeText = "Hello, " + firstName + "I see  
you enjoy " + hobby;} 
```
- **Proprietățile** sunt attribute ce definesc un obiect. De exemplu, _visible este o proprietate a tuturor movie clip-urilor, care definește dacă movie clip-ul este vizibil sau ascuns.

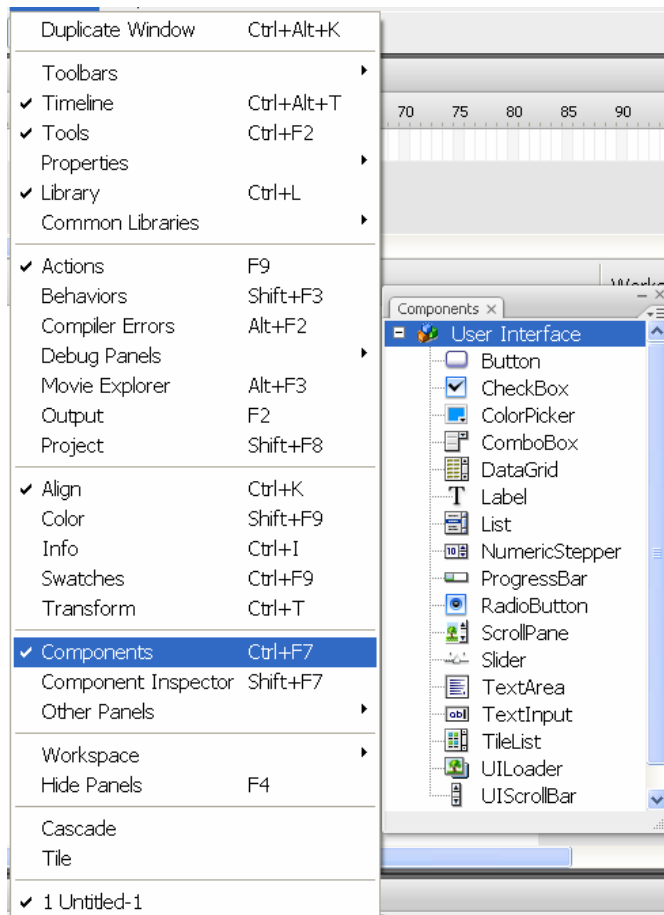


Action Script

- **Target paths** reprezintă o adresă ierahică la numele unei instanțe a unui movie clip, variabile și obiecte dintr-un film (movie). Alocarea unui nume instanței movie clip-ului se face în Property inspector. (Timeline-ul principal are întotdeauna numele `_root`.) Target path se utilizează pentru a iniția o acțiune în movie clip sau pentru a obține sau seta valoarea unei variabile. De exemplu, următorul script este target path la variabila volume din movie clip-ul `stereoControl`:
 - `_root.stereoControl.volume`
 - **Variabilele** sunt identificatori care păstrează valori pentru oricare tip de date. Variabilele pot fi create, modificate și actualizate. Valorile pe care le conțin pot fi utilizate în scripturi. În următorul exemplu, identificatorii de la stânga semnului egal sunt variabile:
 - `x = 5; nume = "Vasile"; client.adresa = "bd.Republicii 123"; c = NColor(mcinstanceName);`

Componente Action Script

• Apelarea componentelor



• Tipuri de componente

- Butoane
- CheckBox
- ColorPicker
- ComboBox
- DataGrid
- Label
- List
- NumericStepper
- ProgressBar
- RadioButton
- ScrolPanel
- Slider
- TextArea
- TextInput
- TileList
- UILoader
- UIScrollBar



Componente Action Script

- Realizarea unei aplicații simple
 - File > New.
 - Flash ActionScript 3.0
 - File > Save welcome fla
 - select **TextArea** și aducerea unei instanțe în scenă
 - În fereastra Properties, având TextArea selectat
 - **aTa** numele instanței
 - **230** pentru W (width).
 - **44** pentru H (height).
 - **165** pentru ordonata X (poziție orizontală).
 - **57** pentru abscisa Y (poziție verticală).
 - **Welcome!** parametru text în tab-ul Parameters.



Componente Action Script

- ColorPicker în stânga TextArea
 - instanța **txtCp**.
 - **96 X**.
 - **72 Y**.
- 3 RadioButton cu instanțele
 - **micRb**
 - **mediuRb**
 - **mareRb**.
 - Property Inspector:
 - **100 W** și **22 H** pentru fiecare.
 - **155 X** pentru toate.
 - **120 Y** pentru micRb, **148** pentru mediuRb și **175** for mareRb.
 - groupName = **fontRbGrp** ca parametru pentru fiecare RB
 - Etichete în tab-ul Parameters cu **Mic, Mediu, Mare**.



Componente Action Script

- ComboBox to the Stage
 - Instanță msgCb
 - **130 W**
 - **265 X**
 - **120 Y.**
- Parametrii
 - **Prompt > Welcome**
 - Dublu click pe dataProvider pentru deschiderea ferestrei Values
 - Click pe semnul + și se înlocuiește cu Bine ai venit!
 - Se repetă adăugând **Încă o dată!** și **Final!**.



Componente Action Script

- Save.
- Se deschide Actions panel cu **F9** sau se selectează Actions din meniul Window
- Se selectează Frame 1 din timeline-ul principal și se introduce următorul cod:

```
/*Se importă clasele de evenimente utilizate*/
```

- import fl.events.ComponentEvent;
- import fl.events.ColorPickerEvent;
- import fl.controls.RadioButtonGroup;

```
/*Este definită variabila rbGrp ca și componentă a  
RadioButtonGroup fiind instanță a fontRbGrp*/
```

- var rbGrp:RadioButtonGroup=
RadioButtonGroup.getGroup ("fontRbGrp");



Componente Action Script

/*Se introduce evenimentul Click pentru rbGrp, txtCp și msgCb prin rbHandler, cpHandler și cbHandler*/

– rbGrp.addEventListener(MouseEvent.CLICK,rbHandler);

– txtCp.addEventListener(ColorPickerEvent.CHANGE,cpHandler);

– msgCb.addEventListener(Event.CHANGE,cbHandler);

/*Se introduce varianta tf pentru modificarea culorii și textului*/

– var tf:TextFormat = new TextFormat();



Componente Action Script

```
/*Definirea modificărilor pentru mărimea fontului pentru butoanele radio*/  
- function rbHandler(event:MouseEvent):void {  
-     switch (event.target.selection.name) {  
-         case "micRb" :  
-             tf.size = 14;  
-             break;  
-         case "mediuRb" :  
-             tf.size = 18;  
-             break;  
-         case "mareRb" :  
-             tf.size = 24;  
-             break;  
-     }  
-     aTa.setStyle("textFormat", tf);  
-     /*aTa numele instanței pentru TextArea*/  
- }
```



Componente Action Script

*/*Se introduce funcția de schimbare a culorii*/*

```
– function cpHandler(event:ColorPickerEvent):void {  
–   tf.color = event.target.selectedColor;  
–   aTa.setStyle("textFormat", tf);  
–   /*aTa numele instanței pentru TextArea*/  
– }
```

*/*Funcția pentru selecția textului din ComboBox*/*

```
– function cbHandler(event:Event):void {  
–   aTa.text = event.target.selectedItem.label;  
– }
```



Butoane Action Script

- Componenta Button este configurabilă ca și dimensiuni, fiind implicit dreptunghiulară putând introduce un eveniment prin clickonmouse sau prin spacebar.
- Se mai poate modifica comportamentul în forma toggle, rămânând apăsat până la următorul click



Butoane Action Script

Crearea evenimentelor pentru un buton
cu AS3.0

- Se creează un nou fișier
- Se aduce o instanță a unui buton cu numele

`aButton` și se introduce la etichetă `Show`

- Se adaugă o instanță `colorPicker` a `Cp`



Butoane Action Script

Crearea butoanelor cu AS3.0

- Se creează un nou fișier
- Se aduce o instanță a unui buton cu numele **aButton** după care se șterge din scenă
- Se introduce codul:
 - `import fl.controls.Button;`
 - `var aButton:Button = new Button(); addChild(aButton);`
 - `aButton.label = "Apasă";`
 - `aButton.toggle = true;`
 - `aButton.move(50, 50);`
 - `aButton.addEventListener(MouseEvent.CLICK, clickHandler);`
 - `function clickHandler(event:MouseEvent):void {`
 - `trace("Event type: " + event.type);`
 - `}`



Referințe

- <http://www.actionscript.org/resources/categories/Tutorials/>
- <http://www.actionscript.org/>
- <http://actionscriptexamples.com/>
- http://www.adobe.com/support/flash/action_scripts/actionscript_tutorial/