

Apostila



Nível Intermediário

Apresentação	(3)
Timeline, Layers e Símbolos	(4)
Modificando o comportamento de um Símbolo	(7)
Exercício	(8)
Ações de controle de reprodução	(9)
Incluindo Ações	(10)
Indicando um Alvo	(11)
Exercício	(12)
ActionScript	(13)
Variáveis	(13)
Texto e Variáveis	(14)
Exercício	(14)
Ações condicionais e laços de repetição	(15)
Exercício	(15)
Propriedades de Objetos	(16)
Exercício	(17)
Carregamento de símbolos e variáveis a partir de arquivos	(18)
Exercício	(18)
Controle de carregamento	(19)
Exercício	(19)

Apresentação

Este curso tem como objetivo apresentar algumas técnicas úteis na criação de conteúdos em Flash. Apresentaremos as principais propriedades dos símbolos e da timeline, e como controlar estas propriedades utilizando comandos ActionScript (códigos de script que permitem programar métodos e procedimentos no Flash).

O método escolhido para o desenvolvimento destes conceitos é a resolução problemas. Durante o curso, serão apresentadas algumas funcionalidades, e em seguida, um problema é proposto. Os alunos devem então tentar chegar a uma solução, desenvolvendo as próprias técnicas para resolução de alguns desafios que são comumente encontrados no desenvolvimento em Flash.

O Flash é apenas uma ferramenta que utilizamos para executar uma idéia.

Timeline, Layers e Símbolos

A Timeline combinada aos layers, forma a estrutura principal de um projeto em flash. É baseado nesta estrutura que devemos organizar os elementos gráficos.

Definimos “o que” e “quando” algo vai aparecer, distribuindo os elementos em layers e definindo em cada layer quadros, quadros-chave ou quadros-chave em branco, de acordo com o objetivo pretendido.

Por exemplo:



Esta é apenas uma das maneiras possíveis de se organizar os elementos da animação.

É possível obter o mesmo resultado utilizando formas diferentes de organização, consegue imaginar como?

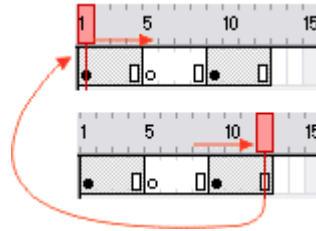
A utilização de símbolos, adiciona uma nova dimensão à estrutura da nossa animação, pois cada símbolo criado possui uma Timeline própria (inclusive com Layers próprios, verifique!), logo, é possível criar animações e estruturas organizadas dentro dos símbolos também.

Exemplo:



Existem três tipos de símbolos, e a principal diferença entre eles é o comportamento de suas Timelines.

O comportamento padrão da Timeline principal é iniciar a reprodução pelo primeiro quadro, e seguir mostrando quadro-a-quadro até atingir o último quadro definido, então ela volta ao primeiro e recomeça a reprodução (loop contínuo).



Característica dos símbolos:

Gráfico (graphic)

- A reprodução da Timeline de um símbolo do tipo “gráfico” (graphic), está vinculada à reprodução da timeline principal. Se a Timeline principal anda um quadro a frente, a Timeline do Símbolo também o faz, se voltamos dois quadros na principal, idem para o Símbolo.

Clipe de Filme (movie clip)

- Tem a Timeline totalmente independente da Timeline principal. Um Símbolo movieclip toca em loop, porém pode ter a reprodução controlada por ações (ActionScript), a partir de qualquer local da animação.

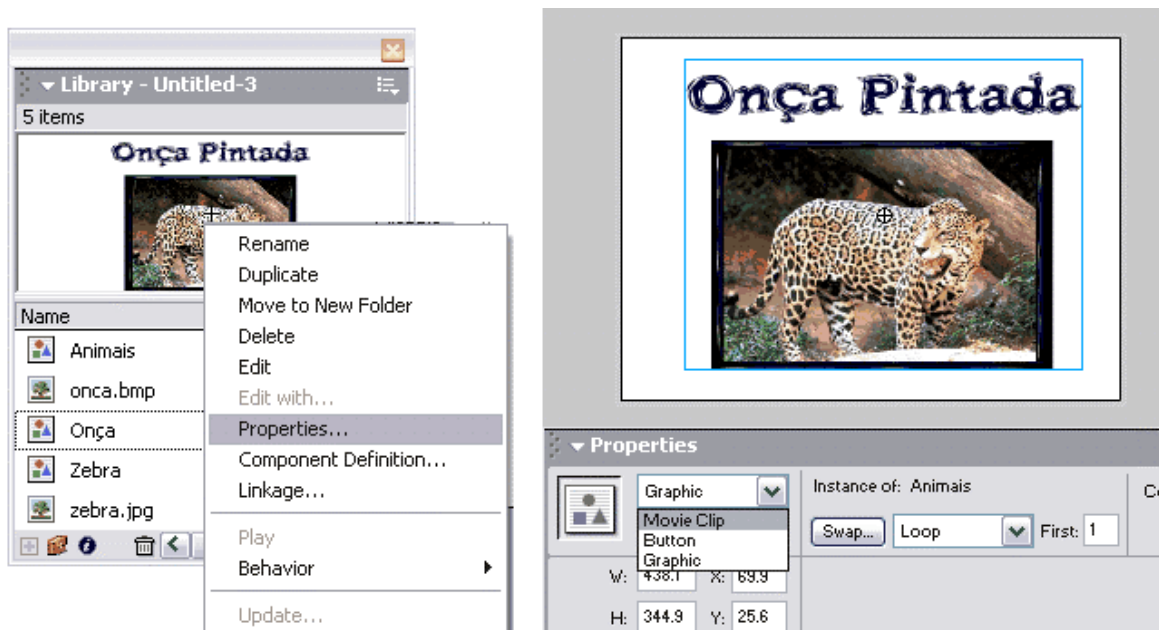
Botão (button)

- A Timeline de um símbolo do tipo “botão” não corre sozinha, somente responde a interações do mouse posicionando-se em um quadro específico (up, over, down) .

Modificando o Comportamento de um símbolo

É possível alterar o tipo de um símbolo depois de criado:

- Selecionando-o na biblioteca com o botão direito do mouse, e alterando suas “Propriedades”, ou
- Selecionando uma instância do símbolo na área de trabalho e alterando o behavior em suas Propriedades.



A segunda maneira só altera o comportamento da instância que está selecionada. O símbolo na biblioteca continuará sendo do tipo que foi criado.

Na “caixa de propriedades” da instância de um símbolo, é possível alterar o vínculo da instância com seu símbolo de origem na biblioteca, clicando em “Swap...” e escolhendo o novo símbolo.

Exercício:

1 – Monte uma animação, apresentando imagens e textos animados, utilizando apenas símbolos do tipo gráfico se necessário.
(exercicio1.swf).

Dicas:

Não existe um método de organização mais correto, o que existe são métodos que de acordo com o objetivo e a lógica aplicada tornam o desenvolvimento e a manutenção da animação menos trabalhosos.

- É possível recortar e colar quadros, ou grupo de quadros, de uma timeline para outra.
- É possível criar um novo símbolo duplicando um símbolo já existente na biblioteca.

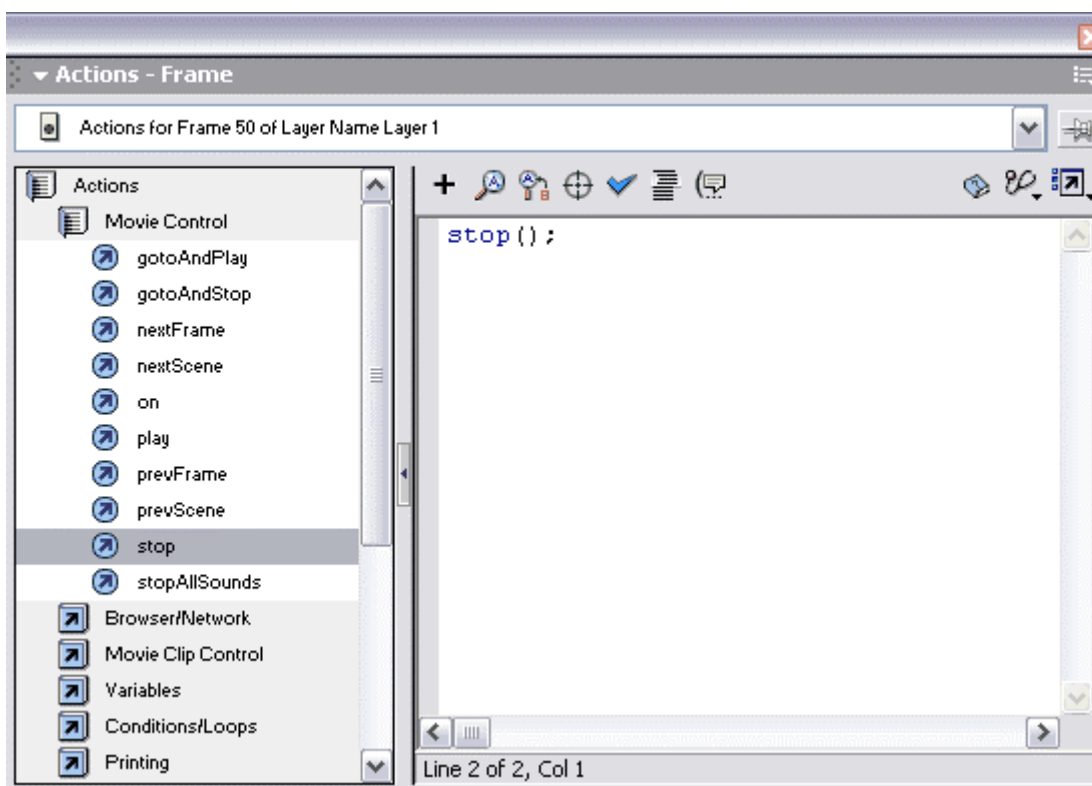
Ações de controle de reprodução

As principais ações para controle de reprodução da animação são:

- play();
- stop();
- nextFrame;
- prevFrame;
- gotoAndPlay();
- gotoAndStop();

Podemos associá-las a quadros na timeline, eventos de botão e eventos de um movieclip.


Para acessar o painel de Ações pressione F9.




dica: Conheça as ações disponíveis navegando pela biblioteca de ações.
(área esquerda da tela)

Incluindo Ações

Para adicionar uma ação, selecione o objeto desejado e na janela de ações, de um duplo clique na ação desejada.

Se preferir digitar os comandos, mude o modo da janela de ações, clique no botão:  e selecione "Expert". Neste modo é preciso saber a sintaxe exata, que com o tempo de uso vai se tornando conhecida.

A qualquer momento é possível verificar se os comandos digitados estão corretos, basta clicar no botão .

Para obter uma descrição da ação selecionada, clique em .

dica: todo comando termina com ponto-e-vírgula.

Indicando um Alvo

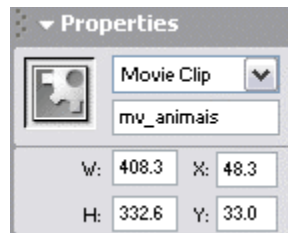
A menos que seja indicado um alvo, a Ação vai atuar na timeline em que foi aplicada. Para indicar um alvo distinto, é preciso primeiro identificá-lo e então adicionar seu nome separado por ponto antes do comando desejado.

A timeline principal é conhecida como Nível 0, para comandar uma ação de qualquer lugar da animação, tendo como alvo a timeline principal, basta digitar “_level0” ou “_root”, antes do comando. Por exemplo:

```
_root.gotoAndStop(5);
```

Quando inserimos um movieclip na timeline principal, ele passa a ser um “segundo nível” de organização, montado em cima do Nível 0 (_root).

Se desejarmos controlar um movieclip, é necessário identificar a instância que se deseja controlar com um nome:



Esta instância está agora acessível de qualquer nível da animação. Para referenciá-la, é preciso identificar sua localização e seu nome.Ex.:
_root.mv_animais

Para parar este movieclip de qualquer ponto da animação, adicionamos a ação:

```
_root.mv_animais.stop();
```

Exercícios:

2 - Modifique o arquivo do exercício 1, utilizando agora, apenas um quadro da timeline principal para cada figura.

3 - Construa um “timer”, para controlar o tempo de troca entre cada figura.

4 - Construa um menu de navegação para a animação.

Dica: É possível utilizar várias instâncias de um mesmo botão transparente, para executar ações diferentes.

ActionScript

Através dos códigos ActionScript, é possível se fazer muito mais do que apenas controlar a reprodução do filme. Em alguns casos, toda uma aplicação pode ser gerada via ActionScript. O mais comum no entanto, é utilizar a linguagem de script apenas como suporte, para o que não conseguimos fazer visualmente.

Na versão atual do Flash (MX), a linguagem de script alcançou uma certa “maturidade”, possui todos os principais recursos de uma linguagem de programação e acessa/modifica todas as propriedades dos elementos criados visualmente.

Variáveis

Variáveis são como repositórios onde podemos guardar informações úteis, para manipular e utilizar mais tarde, como contadores, nome de um usuário, etc...

Para criar variáveis, basta associar um nome a um valor. por exemplo:

```
var item = 3;  
var nome = 'Jacaré';
```

Alguns tipos de variáveis especiais, precisam ser inicializados, é o caso de variáveis do tipo array, que são listas numeradas:

```
lista = new array();  
lista[1] = "maçã";  
lista[2] = "banana";  
lista[3] = "pera";
```

O comando trace() pode ser utilizado para mostrar o valor de uma variável.

```
trace(_root.item);
```

As variáveis são criadas no nível em que são declaradas. Para manipular uma variável declarada em um outro nível, é preciso indicar sua localização.

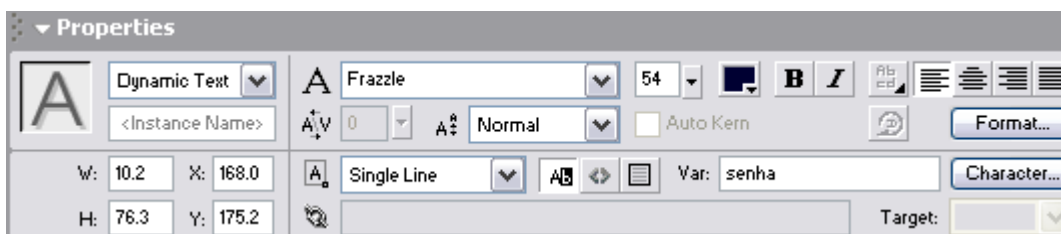
Por exemplo:

```
trace(_root.mv_animais.nome);  
trace(_root.lista[3]);
```

Textos e variáveis

Elementos do tipo texto podem estar vinculados a variáveis. Para tal, é preciso mudar o tipo da caixa de texto, no painel “propriedades do texto”.

O padrão é Texto Estático (static text), mude para Texto Dinâmico (dynamic text) se deseja apenas mostrar um texto ou Entrada de Texto(input text) se deseja que o usuário digite em uma caixa de entrada.



Exercícios:

5 – Crie uma “máquina de animações de texto” (exercício5.swf)

Ações condicionais e laços de repetição

Em algumas animações, é preciso se verificar uma condição antes de prosseguir (uma tela de Login por exemplo). Nestes casos podemos utilizar o comando "if" para verificar uma condição. Exemplo:

```
on(release)
{
  if ( pass == 'senha' ) { gotoAndStop(2);}
  else { gotoAndStop(3); }
}
```

De maneira semelhante definimos os laços de repetição, que permitem repetir um bloco de procedimentos.

```
for (x=1; x<4; x++)
{ trace(_root.lista[x]); }
-----
x=1;
while (x < 4 )
{ trace(_root.lista[x]);
  x++; }
```

Ambos os códigos produzem o mesmo resultado: mostram os valores da variável `_root.lista[]`, nas posições de 1 a 4.

Tome nota:

- Utilizamos chaves {} para indicar o início e o fim de um bloco de execução
- Para comparações de valor, devemos utilizar um Duplo sinal de igual.

Exercício

6 - Crie uma tela de login, que peça nome e senha, libere o acesso no caso dos campos estarem certos, e trave depois de 3 tentativas erradas.

Propriedades de Objetos

Através de ActionScript, é possível acessar e alterar a propriedade de alguns objetos da animação como botões, movieclips e caixas-de-texto. Algumas propriedades úteis são:

<code>_x</code>	posição horizontal	pixels
<code>_y</code>	posição vertical	pixels
<code>_width</code>	largura	pixels
<code>_height</code>	altura	pixels
<code>_visible</code>	visível	true/false
<code>_rotation</code>	rotação	1 - 360

Outras propriedades estão disponíveis na biblioteca de ações, e podem ser acessadas de maneira semelhante;

Exemplo de uso :

Este código inverte as coordenadas de posição de um movieclip

```
{  
pos_y= _root.movieclip._y;  
pos_x= _root.movieclip._x;  
_root.movieclip._x = pos_y;  
_root.movieclip._y = pos_x;  
}
```

Exercício

7 – Monte um mapa navegável (exercicio7.swf)



Extra: Que tal adicionar uma ferramenta de zoom para o mapa ?

Carregando símbolos e variáveis a partir de arquivos

O comando *loadMovie()* permite o carregamento de um arquivo externo em um símbolo dinâmico na animação, que pode ser manipulado como um símbolo nativo. Utilização:

```
loadMovie('nome do arquivo', 'nome de destino');
```

Para descarregar o arquivo da animação utilizamos o comando *unloadMovie()*:

De maneira semelhante, o comando *loadVariables()* possibilita o carregamento de variáveis e respectivos valores previamente declarados em um arquivo de texto. Este recurso é muito útil em sites que possuem atualização rotineira de conteúdo. Utilização:

```
loadVariables('arquivo texto', 'nível da animação');
```

Exercício:

8 – Crie um site em que os textos dos menus são carregados dinamicamente. Cada item do menu deve carregar um arquivo de animação (exercício8.swf)

Controle de Carregamento

A medida em que adicionamos conteúdo em nossas animações, cresce o tamanho do arquivo final. Quando o destino é a internet, a animação pode demorar algum tempo para carregar.

Nestes casos, podemos criar indicadores de carregamento, que indicam ao usuário quanto falta a carregar, para tal, é preciso programar uma pequena rotina, que testa quantos quadros foram carregados do total, utilizando as propriedades e métodos:

- *_framesloaded*
- *_totalframes*
- *getBytesLoaded()*
- *getBytesTotal()*.

Exercício:

9 – Monte esta rotina, tendo como exemplo o arquivo (exercicio9.swf)