

FLASH **WEBTRAINING.COM** **TUTORIALS**



VISUALWORK

DYNAMIC MENU

Adicionar som a um filme de Flash é bastante útil quando queremos que os nossos visitantes não se sintam aborrecidos, enquanto navegam num complexo interface criado com a ajuda de 'actionsript'. Mas adicionar som, pode não ser tarefa fácil. Somente importar um som para a biblioteca e de seguida começar a reproduzir o mesmo som com AS não é fácil. O nosso visitante pode facilmente ficar saturado se não lhe dermos acesso ao controle de volume.

1. Antes de tudo, deveremos tomar conhecimento que o site contém duas **scenes**. Na primeira **scene** está contido um preloader e na segunda **scene** está contida a homepage. Por isso mesmo, criem um novo ficheiro de Flash e adicionem uma nova **scene**. Este é o primeiro passo a ser dado. A partir deste momento, iremos trabalhar na segunda **scene**.
2. Agora, necessitamos de importar um som para a biblioteca. Para tal poderemos pressionar as teclas de CTRL+R e seleccionar o ficheiro de som pretendido.
3. Após termos completado o passo 2, deveremos abrir a biblioteca pressionando a tecla de F11, localizar o ficheiro de som, clicar com o botão direito do rato sobre o ficheiro e escolher a opção de **Linkage**. Neste tutorial usaremos o nome de 'backsound' para este ficheiro de som.
4. Agora que o ficheiro de som foi exportado para AS, continuaremos com a magia. Dirijam-se à primeira frame, abram o painel de action script e iniciem o som. Para tal deveremos inserir o seguinte código:

```
music = new Sound();
music.attachSound("backsound");
music.start(0, 999999);
```

5. Como não queremos que o som comece abruptamente, deveremos programá-lo de forma a que comece suavemente. Para tal, primeiro deveremos iniciar o valor de início do volume para 0.

```
music.setVolume(0);
```

6. Neste momento necessitamos de criar uma função que suavize o som desde o seu início. Para que a função funcione necessitamos de definir um conjunto de variáveis. A nossa função inicial de **fade in** será assim elaborada:

```
vol = 0;
fade = setInterval(fadeIn, 100);
function fadeIn() {
    vol += 3;
    music.setVolume(vol);
    if (vol >= 100) {
        clearInterval(fade);
    }
}
```

7. O que acontece aqui é que esta função cria um **fade** dinâmico com um aumento de três ao volume que está posicionado no nível zero. É de referir que o valor de incremento pode ser mudado consoante as nossas necessidades. Quando o volume chegar ao valor de 100, o intervalo é estagnado. Agora teremos de criar uma nova função, para que o som possa fazer um **fade in** e **fade out** de acordo com os cliques efectuados no botão. Desta forma teremos de incluir o seguinte código no painel das Actions:

```
_root.onEnterFrame = function() {
    if (Fade == 1) {
        vol = vol-step;
        if (vol < 0) {
            vol = 0;
        }
        music.setVolume(vol);
    } else {
        vol = vol+step;
        if (vol > 100) {
            vol = 100;
        }
        music.setVolume(vol);
    }
}
```

```
};
```

8. Seguramente que está algo a faltar ainda. Poderemos ver perfeitamente que as variáveis para o nosso **fade in** e **fade out** são o 'fade' e 'step'. O Step é valor que queremos que o som faça **fade in** e **fade out** e a variável 'fade' é a variável que diz que a função que deve fazer um **fade in** ou **fade out** consoante os cliques no botão. Para tal deveremos voltar à nossa função inicial do **fade** e adicionar duas linhas para definir as variáveis. Finalmente, a nossa primeira função irá parecer-se com o seguinte:

```
vol = 0;
fade = setInterval(fadeIn, 100);
function fadeIn() {
    vol += 3;
    music.setVolume(vol);
    if (vol>=100) {
        clearInterval(fade);
        step = 1;
        Fade = 0;
    }
}
```

9. Assim sendo, teremos o corpo do nosso código definido na sua totalidade desta maneira:

```
music = new Sound();
music.attachSound("backsound");
music.start(0, 999999);
music.setVolume(0);
vol = 0;
fade = setInterval(fadeIn, 100);
function fadeIn() {
    vol += 3;
    music.setVolume(vol);
    if (vol>=100) {
        clearInterval(fade);
        step = 1;
        Fade = 0;
    }
}
_root.onEnterFrame = function() {
    if (Fade == 1) {
        vol = vol-step;
        if (vol<0) {
            vol = 0;
        }
        music.setVolume(vol);
    } else {
        vol = vol+step;
        if (vol>100) {
            vol = 100;
        }
        music.setVolume(vol);
    }
};
```

10. Finalmente, deveremos criar um botão, ao qual deverá ser acrescentado o seguinte código de ActionScript:

```
on (release) {
    (_root.fade=!_root.fade) ? 0 : 1;
}
```

O que acontece é que quando clicamos pela primeira vez no botão, é atribuído o valor de '1' à variável. O valor da variável 'fade' é posto no valor de 0 na sua primeira função. Quando se clica no botão pela segunda vez este funciona da maneira contrária. O comando ?0:1 simplesmente atribui o valor da variável 'fade' para o valor de 1 ou 0, dependendo do seu valor prévio.

11. Testem o vosso projecto.

Esperamos que este tutorial, vos tenha sido de alguma utilidade. Se tiveres alguma dúvida ou questão, não hesites em utilizar os nossos Fóruns de Discussão.

Bons trabalhos em Flash.
Pedro Teixeira