

Introduction

La gestion du son en AS3 est logique si on se réfère au fonctionnement d'une table de mixage audio !

En AS 1 et 2, il suffisait d'instancier la classe `Sound()` et nous avions alors accès à toutes ses méthodes et propriétés. Pour gérer plusieurs sons en même temps, il fallait savoir qu'il était nécessaire d'instancier la classe `Sound()` en passant le nom d'une occurrence entre les parenthèses de la classe `Sound()`. Aujourd'hui, le principe de lecture et de contrôle d'un son passe par un minimum de 3 classes : `Sound()`, `SoundChannel()` et `SoundTransform()`. Cette "complexité" est une fois de plus liée à l'architecture du langage.

Voici un exemple d'animation qui gère la lecture de deux sons en cliquant sur deux occurrences de `Clip`, vous pouvez mettre la lecture en pause, cliquer sur la jauge de lecture et régler le niveau sonore (Commencez par cliquer sur l'un des deux boutons `Piano1` ou `Piano2`) :

[\[Télécharger le .fla\]](#)

Remarque : Il est normal que la jauge qui indique la part de lecture "s'emballe" au début de l'écoute, il est en effet impossible de connaître la durée d'un son tant qu'il n'est pas complètement chargé. Pour résoudre le problème, il faudrait lire cette information dans un tag du fichier mp3.

Avant de nous lancer dans l'apprentissage de la gestion du son en AS3, essayons de

comprendre le principe suivant :

D'un point de vue professionnel, lorsque vous lancez l'écoute d'un son, cela se fait sur une piste précise (voir l'image de la table de mixage ci-dessous). Vous associez une piste à un son pour pouvoir le contrôler indépendamment des autres (même si vous ne gérez qu'un seul son), en ActionScript, nous devons donc reproduire le même "schéma" !

Pour effectuer vos premiers essais, vous pouvez utiliser les fichiers ci-dessous qui sont très courts et libres de droits (effectuez un clic-droit sur chaque lien et sélectionnez la commande Enregistrer la cible sous...) :

[Son de Piano N°1](#) [Son de Piano N°2](#) [Son de Piano N°3](#) [Son de Piano N°4](#)

Lire un son □

Pour pouvoir écouter un son dans une animation Flash, vous devez "créer un nouveau son" et le lire, en le chargeant préalablement. Pour ce faire, vous devez instancier les classes Sound() et URLRequest().

```
var ecoute:Sound = new Sound();
```

```
var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");
```

```
ecoute.load(adresseFichierSon);
```

Remarque : L'adresse du fichier que vous indiquez entre les parenthèses de la classe `URLRequest()` est relative à la page HTML qui contient le SWF.

Lorsque vous avez saisi ces 3 lignes d'instructions, vous pouvez ensuite demander la lecture du son :

```
ecoute.play();
```

Remarque : Si le script ci-dessus vous semble un peu complexe, vous pouvez vous dispenser du typage.

```
var ecoute = new Sound();
```

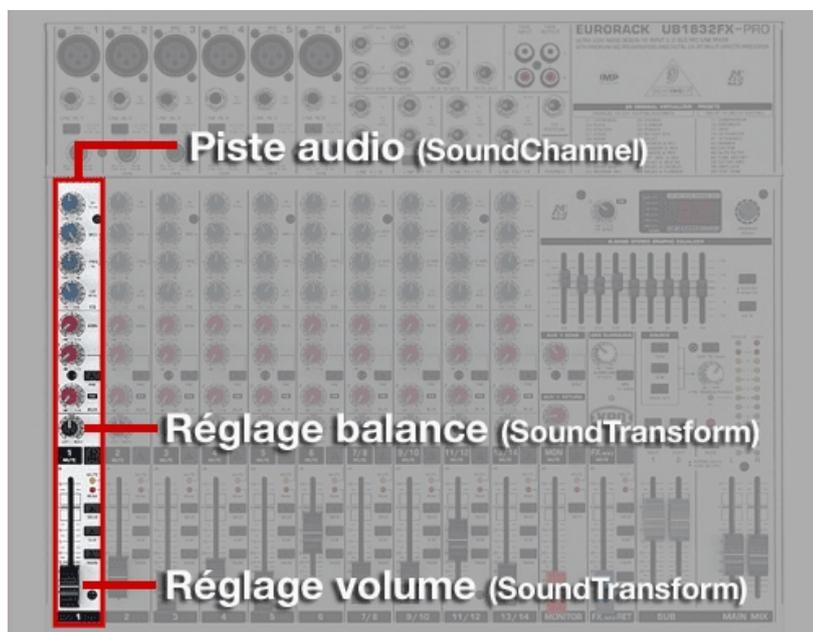
```
var adresseFichierSon = new URLRequest("Piano1.mp3");
```

```
ecoute.load(adresseFichierSon);
```

```
ecoute.play();
```

[\[Télécharger le .fla\]](#)

Avant de poursuivre nos explications, voici une photo qui représente une table de mixage audio. Observez attentivement les 3 textes qui y sont associés avant de poursuivre la lecture de cette ressource.



Contrôler un son

Lorsque vous allez vouloir contrôler la lecture et l'arrêt d'un son, vous rencontrerez peut-être un premier problème si vous n'avez pas gardé à l'esprit qu'il est indispensable d'associer la lecture d'un son à une piste. Voici donc le script à utiliser dès que vous aurez placé sur la scène, deux occurrences de type Clip intitulées **boutonLecture** et **boutonArrêt**.

```
var ecoute:Sound = new Sound();

var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");

var piste1:SoundChannel;

ecoute.load(adresseFichierSon);

boutonLecture.addEventListener(MouseEvent.CLICK,lireSon);

boutonArret.addEventListener(MouseEvent.CLICK,arreterSon);

function lireSon(evt:MouseEvent) {

    piste1 = ecoute.play();

}

function arreterSon(evt:MouseEvent) {

    piste1.stop();

}
```

Lorsque vous exécuterez le script ci-dessus, si vous double-cliquez sur le bouton Lecture, vous découvrirez peut-être que vous entendez de l'écho. Cela est dû au fait que le son se lance deux fois. Si vous effectuez d'ailleurs 3 clics au lieu de 2, vous entendez 3 fois le son se jouer. Pour éviter ce bug, vous devez adapter votre script de la façon suivante :

```
var ecoute:Sound = new Sound();

var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");

var piste1:SoundChannel;

ecoute.load(adresseFichierSon);

boutonLecture.addEventListener(MouseEvent.CLICK,lireSon);

boutonArret.addEventListener(MouseEvent.CLICK,arreterSon);

function lireSon(evt:MouseEvent) {

    piste1 = ecoute.play();

    boutonLecture.enabled = false;
```

```
}  
  
function arreterSon(evt:MouseEvent) {  
  
    piste1.stop();  
  
    boutonLecture.mouseEnabled = true;  
  
}
```

[\[Télécharger le .fla\]](#)

Comme vous pouvez le constater, nous avons ajouté la ligne d'instruction qui permet d'annuler la reconnaissance du clic sur l'occurrence boutonLecture, mais nous devons penser à rétablir l'état cliquable du bouton pour pouvoir relancer la lecture du son.

Pour donner l'impression que le bouton Lecture n'est pas actif, vous pouvez utiliser la propriété alpha et la régler à 0.5 pour rendre l'occurrence légèrement transparente (Pensez à la rendre à nouveau opaque lorsque vous arrêtez le son et que le son s'arrête automatiquement). Consultez la ressource liée aux filtres pour appliquer et retirer éventuellement une ombre portée au bouton Lecture en fonction de l'écoute d'un son.

Remarque : Si vous attendez la fin de la lecture du son, vous constaterez que vous ne pouvez pas relancer le son. C'est normal, vous devez gérer les actions à exécuter à la fin de la lecture d'un son. Consultez le développement "Gérer la fin de la lecture d'un son" plus loin dans cette page.

Pour arrêter un son, si les propos ci-dessus vous semblent trop abstraits, et si vous ne gérez qu'un seul son à la fois, vous pouvez dans ce cas utiliser la ligne d'instruction ci-dessous :

```
SoundMixer.stopAll()
```

Contrôler le démarrage du son et le nombre de répétitions (boucles)

Pour démarrer la lecture d'un son à partir d'un instant précis, indiquez entre les parenthèses de la méthode `play()`, le nombre de millisecondes à partir desquelles la lecture doit commencer.

```
piste1 = ecoute.play(2000,1);
```

Dans cet exemple, la lecture démarre à la deuxième seconde (à partir de la 2000^e milliseconde, soit l'équivalent de 2 secondes). Pour lire un son plusieurs fois, vous devez préciser le nombre de répétitions entre les parenthèses de la méthode `play()`, au niveau du deuxième paramètre. Dans l'exemple qui va suivre, la lecture du son va s'exécuter 5 fois, à partir du début.

```
piste1 = ecoute.play(0,5);
```

Nous pourrions également combiner les deux paramètres que nous venons de voir, dans l'exemple ci-dessous, un son va être joué 3 fois à partir de la 7^e seconde :

```
piste1 = ecoute.play(7000,3);
```

Gérer la fin de la lecture d'un son

Lorsqu'un son se termine, il est très important de le savoir pour pouvoir agir en conséquence : (ex. : Enchaîner un autre son, afficher ou masquer une occurrence, rendre le bouton lecture à nouveau cliquable (s'il a préalablement été rendu non-cliquable avec la propriété `mouseEnabled`), etc.). Voici l'exemple d'un script où le bouton Lecture redevient cliquable :

```
piste1.addEventListener(Event.SOUND_COMPLETE,ecouteSonTerminee);
```

```
function ecouteSonTerminee(evt:Event) {
```

```
    trace("Fin du son");
```

```
}
```

Vous devez enregistrer un écouteur qui fait appel à l'évènement `Event.SOUND_COMPLETE`, et faire référence à l'instance de la classe `SoundChannel()`, c'est à dire `piste1` dans notre exemple. Voici un exemple de script complet :

```
var ecoute:Sound = new Sound();
```

```
var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");
```

```
var piste1:SoundChannel;
```

```
ecoute.load(adresseFichierSon);
```

```
boutonLecture.addEventListener(MouseEvent.CLICK,lireSon);
```

```
function lireSon(evt:MouseEvent) {
```

```
    piste1 = ecoute.play();
```

```
    boutonLecture.enabled = false;
```

```
    piste1.addEventListener(Event.SOUND_COMPLETE,ecouteSonTerminee);
```

```
}
```

```
function ecouteSonTerminee(evt:Event) {
```

```
    boutonLecture.enabled = true;
```

```
}
```

[\[Télécharger le .fla\]](#)

Effectuer une pause lors de l'écoute d'un son

L'une des difficultés dans le développement d'une fonctionnalité de pause pour un lecteur de fichier MP3 est relative à la réalisation du bouton pause en lui-même. Pour faire simple, vous pouvez donc créer un clip avec deux images clé. Sur la première, vous créez l'apparence d'un bouton pause, sur la deuxième, vous créez l'apparence d'un bouton de lecture (pensez à placer un `stop()` sur la première image clé de ce clip). Dans la mesure où la méthode `pause()` n'existe pas, nous devons donc la programmer en faisant appel à une variable qui va mémoriser la position de la tête de lecture au moment de la pause.

Voici un exemple d'animation contenant un bouton pause qui autorise la lecture et la pause d'un son à partir d'un même bouton :

```
var ecoute:Sound = new Sound();
```

```
var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");
```

```
var piste1:SoundChannel;
```

```
var positionTeteMemorisee:Number = 0;
```

```
ecoute.load(adresseFichierSon);
```

```
piste1 = ecoute.play();
```

```
btLecturePause.addEventListener(MouseEvent.CLICK,ecouterSon);
```

```
function ecouterSon(evt:MouseEvent) {
```

```
    if (btLecturePause.currentFrame==1) {
```

```
        positionTeteMemorisee=piste1.position;
```

```
        piste1.stop();
```

```
        btLecturePause.gotoAndStop(2);
```

```
    } else {
```

```
        piste1 = ecoute.play(positionTeteMemorisee);
```

```
        btLecturePause.gotoAndStop(1);
```

```
    }
```

```
}
```

[\[Télécharger le .fla\]](#)

Gérer le niveau du volume sonore

Souvenez-vous de ce que nous vous avons dit au début de cette ressource, au sujet de la lecture d'un son. Elle doit (se faire) être associée à une piste. A partir de ce postulat, vous comprendrez mieux les lignes d'instructions ci-dessous :

```
var reglagesAudio:SoundTransform = piste1.soundTransform;
```

```
btNiveau100.addEventListener(MouseEvent.CLICK,reglerSona100)
```

```
btNiveau50.addEventListener(MouseEvent.CLICK,reglerSona50)
```

```
btNiveau10.addEventListener(MouseEvent.CLICK,reglerSona10)
```

```
function reglerSona100 (evt:MouseEvent) {
```

```
    reglagesAudio.volume = 1;
```

```
    piste1.soundTransform = reglagesAudio;
```

```
}
```

```
function reglerSona50 (evt:MouseEvent) {
```

```
    reglagesAudio.volume = 0.5;
```

```
    piste1.soundTransform = reglagesAudio;
```

```
}
```

```
function reglerSona10 (evt:MouseEvent) {
```

```
    reglagesAudio.volume = 0.1;
```

```
    piste1.soundTransform = reglagesAudio;
```

```
}
```

[\[Télécharger le .fla\]](#)

Nous déclarons (avec le mot var) et initialisons (en utilisant l'opérateur=) une variable (une instance de type **SoundTransform**) intitulée **reglagesAudio** qui stocke toutes les informations

actuelles relatives au volume et à la balance du son. L'instance

reglagesAudio

possède les propriétés

volume

et

pan

qui permettent de régler respectivement le volume et la balance d'un son. Après avoir modifié la valeur de la propriété

volume

de l'instance

reglagesAudio

, nous devons appliquer le changement en exécutant la ligne

piste1.soundTransform = reglagesAudio

.

Vous l'aurez peut-être compris, pour régler le volume d'un son à 100%, vous devez utiliser la valeur 1, pour le régler à 10%, vous utiliserez la valeur 0.1. Attention, vous pouvez aller au delà de 1, le son sera plus fort, mais vous ne devez pas dépasser 4 car le son ne sera plus de bonne qualité.

Remarques

- Vous ne devez déclarer votre instance de type **SoundTransform** qu'à partir du moment où vous avez défini l'instance de type

SoundChannel

. Ainsi, dans le cas où vous initialiseriez votre instance de type

SoundChannel

dans une fonction, votre instance de type

SoundTransform

doit l'être également à ce moment-là.

- Pour pouvoir augmenter ou diminuer le son à partir de deux boutons (un bouton pour augmenter le son, un autre pour le diminuer), vous devez utiliser les lignes d'instructions ci-dessous en remplacement de celles qui se trouvent dans les fonctions `reglerSona10`, `reglerSona100` et `reglerSona100` ci-dessus.

```
reglagesAudio.volume += 0.1;
```

```
piste1.soundTransform = reglagesAudio;
```

Gérer la balance d'un son

Consultez le développement relatif au réglage du niveau sonore d'un son en utilisant la propriété **pan** en remplacement de **volume**. Les valeurs à utiliser vont de -1 (Gauche) à 1 (Droite).

Afficher le temps écoulé sous la forme 00:00:00

Vous aurez parfois besoin d'afficher le temps qui s'écoule lors de l'écoute d'une musique ou une chanson. Commencez par placer sur la scène de votre animation, un texte dynamique, nommez cette occurrence "affichageTemps". Utilisez alors le code ci-dessous :

```
var ecoute:Sound = new Sound();
```

```
var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");
```

```
var piste1:SoundChannel;
```

```
var tempsEcoule:Number;
```

```
var heures:*;
```

```
var minutes:*
```

```
var secondes:*
```

```
ecoute.load(adresseFichierSon);
```

```
boutonLecture.addEventListener(MouseEvent.CLICK,lireSon);
```

```
function lireSon(evt:MouseEvent) {
```

```
    piste1 = ecoute.play();
```

```
    boutonLecture.enabled = false;
```

```
    piste1.addEventListener(Event.SOUND_COMPLETE,ecouteSonTerminee);
```

```
    boutonLecture.addEventListener(Event.CLICK,afficherTemps);
```

```
}
```

```
function ecouteSonTerminee(evt:Event) {
```

```
    boutonLecture.enabled = true;
```

```
boutonLecture.removeEventListener(Event.ENTER_FRAME,afficherTemps);

}

function afficherTemps(evt:Event) {

    tempsEcoule = piste1.position/1000;

    heures = Math.floor(tempsEcoule/3600);

    minutes = Math.floor(tempsEcoule/60);

    secondes = Math.floor(tempsEcoule%60);

    heures= heures<=9 ? "0"+heures : heures;

    minutes= minutes<=9 ? "0"+minutes : minutes;

    secondes= secondes<=9 ? "0"+secondes : secondes;

    affichageTemps.text = heures+":"+minutes+":"+secondes;

}
```

[\[Télécharger le .fla\]](#)

Nous utilisons la propriété **position** de la classe **SoundChannel** qui nous renvoie le nombre de millisecondes à laquelle se trouve la tête de lecture. Nous utilisons également un gestionnaire d'évènement de type

Event.ENTER_FRAME

pour pouvoir exécuter une action en continue. Pour finir, nous devons calculer le temps écoulé en secondes et minutes (nous avons également prévu les heures dans notre exemple). Comme vous le montre cet exemple, vous devez penser à supprimer l'écouteur qui gère l'évènement

Event.ENTER_FRAME

Réaliser une jauge qui indique le pourcentage de lecture

Si vous n'avez pas encore consulté de développement précédent, faites le avant de poursuivre la lecture sur le script qui permet d'afficher une jauge de lecture. Pour réaliser cette dernière, nous nous sommes basés sur le même code, mais nous avons fait référence en plus à la propriété `length` de la classe `Sound()`. Il est à noter que vous ne pouvez pas avoir une information complète sur la durée d'un son tant qu'il n'est pas chargé complètement.

```
var ecoute:Sound = new Sound();
```

```
var adresseFichierSon:URLRequest = new URLRequest("Piano1.mp3");
```

```
var piste1:SoundChannel;
```

```
ecoute.load(adresseFichierSon);
```

```
boutonLecture.addEventListener(MouseEvent.CLICK,lireSon);
```

```
function lireSon(evt:MouseEvent) {
```

```
    piste1 = ecoute.play();
```

```
    boutonLecture.mouseEnabled = false;
```

```
    piste1.addEventListener(Event.SOUND_COMPLETE,ecouteSonTerminee);
```

```
    boutonLecture.addEventListener(Event.CLICK,animerJauge);
```

```
}
```

```
function ecouteSonTerminee(evt:Event) {
```

```
    boutonLecture.mouseEnabled = true;
```

```
    boutonLecture.removeEventListener(Event.CLICK,animerJauge);
```

```
}
```

```
function animerJauge(evt:Event) {  
  
    if (piste1.position/ecoute.length>0) {  
  
        jaugeProgression.scaleX = piste1.position/ecoute.length;  
  
    }  
  
}
```

[\[Télécharger le .fla\]](#)

Voilà, nous avons abordé toutes les notions élémentaires pour une bonne maîtrise du son dans une animation Flash, il ne vous reste plus qu'essayer de combiner le tout pour réaliser un "player MP3". Si vous savez déjà gérer le XML et le chargement d'une image en AS3, voici un exemple de player qui gère toutes les fonctionnalités que nous avons abordées. Consultez éventuellement les ressources liées au chargement d'une image et la gestion du XML dans Flash pour mieux comprendre cette animation.

[\[Télécharger le .fla\]](#)