Contrôler la tête de lecture d'une animation AS 3.0

1 Arrêter la tête de lecture : stop ()
2 Libérer la tête de lecture : play ()
3 Déplacement vers une image : gotoAnd…()
4 Déplacement vers l'image suivante : nextFrame
5 Déplacement vers l'image précédente : prevFrame()
6 Contrôler la tête de lecture d'une occurrence
7 Appeler une page HTML depuis une animation

Avant-propos

Avec l'apprentissage des instructions qui vont suivre, vous aurez ensuite la possibilité de réaliser un "site" complet en Flash. Attention, modérons notre propos en ajoutant que vous allez pouvoir réaliser une animation comparable à une navigation de pages en pages (HTML). Dans les exemples suivants, nous allons voir les techniques qui permettent de changer le contenu de la scène, c'est à dire changer ce que vous voyez dans votre animation en utilisant de simples commandes.

Rappels :

- Une animation est une suite d'images présentées dans un scénario sur laquelle se déplace la tête de lecture.
- Lorsque vous lancez votre animation, si aucune instruction n'est exécutée, la lecture de votre animation se fera en boucle, et sans arrêt de la tête de lecture, de la première à la dernière image.
- Pour insérer une commande de script avec le raccourci clavier combinant la touche Escape (Echappe) qui se trouve en haut à gauche de votre clavier, pensez à ne pas maintenir cette touche comme vous le faites habituellement avec la touche CTRL. Appuyez une fois sur la touche Esc, relâchez la, appuyez sur une première touche, relâchez la, appuyez enfin sur une deuxième touche.

1 Arrêter la tête de lecture : stop ()

Cette commande permet d'arrêter (bloquer) la tête de lecture lorsque l'animation est en cours de lecture. Elle peut-être placée sur une image-clé, arrêt obligatoire, ou dans un gestionnaire (de bouton), arrêt conditionnel.

1.1 Arrêt obligatoire

Dans l'exemple du fichier **Exercice_8.swf**, l'arrêt de la tête de lecture se fait automatiquement.

- Exécutez le fichier **Exercice_7.swf** : la lecture se fait en boucle.
- Exécutez le fichier **Exercice_8.swf** : la lecture se fait jusqu'à l'image 20, la tête de lecture reste bloquée sur cette image.
- Ouvrez le fichier **Exercice_7.fla** : la lecture se fait en boucle, il n'y a aucun code AS dans cette animation.
- insérez sur une image clé en image 20 du scénario, raccourci clavier : **F6**
- Placez l'instruction ci-dessous dans la fenêtre Actions de l'image 20.

```
1  stop() ;
```

- Testez votre animation (raccourci **CTRL + ENTREE**) : le texte ne défile plus complètement … C'est bien, cette première commande AS 3.0 fonctionne !
- Enregistrez votre animation sous **Exercice_8.fla** dans votre répertoire personnel.

1.2 Arrêt conditionnel

- Ré-ouvrez le fichier **Exercice_7.fla** (si le fichier a changé, enregistrement malheureux, recopiez-le depuis \\KAPOUER\SHS\PUB\Flash **ou du blog**)
- Sélectionnez l'occurrence du bouton *boutonStop* placée sur le calque « Bouton », puis nommez la : *btArret_btn*
- Placez le script ci-dessous sur l'image clé 1 du calque Actions:

```
1  Object(root).btArret_brn.addEventListener(MouseEvent.CLICK, arretLecture);

2  function arretLecture(evt:MouseEvent) {
3      Object(root).stop();
```

Contrôler la tête de lecture d'une animation AS 3.0

```
4   }
5
```

Remarque : Pour insérer rapidement `function() {}` vous pouvez aussi utiliser le raccourci **ESC, F, N.**

- Enregistrez votre animation en **Exercice_9.fla**
- Testez votre animation …

Vous pouvez stopper la lecture en appuyant sur le bouton rouge !
Et pour reprendre la lecture …

2 Libérer la tête de lecture : play ()

Cette commande permet de relancer la tête de lecture lorsqu'elle a été préalablement "stoppée" avec la commande `stop()`.

- Exécutez le fichier **Exercice_10.swf** :

la lecture se fait en boucle, le bouton rouge sert à arrêter la lecture, le vert à la relancer.

- Ouvrez le fichier **Exercice_10.fla** : le bouton Stop est déjà « programmé »
- Sélectionnez l'occurrence du bouton *boutonLecture* puis nommez-la *btLecture_btn*
- Placez le script ci-dessous sur l'image clé 1 du calque Actions, à la suite du code permettant de gérer le bouton btArret :

```
5   Object(root).btLecture_btn.addEventListener(MouseEvent.CLICK,
    repriseLecture);

6   function repriseLecture(evt:MouseEvent) {
7       Object(root).play();
8   }
9
```

- Enregistrez votre animation
- Testez la…

Le bouton rouge arrête la lecture, le vert la reprend. Félicitations !

3 Déplacement vers une image : gotoAnd…()

Il existe deux commandes AS permettant de passer directement la tête de lecture du scénario sur une image précise :

- ⇨ gotoAndStop(…) : déplace la tête de lecture à l'image dont le numéro est placé entre les parenthèses, et y bloque la tête de lecture
- ⇨ gotoAndPlay(…) : déplace la tête de lecture à l'image dont le numéro est placé entre les parenthèses, et libère la tête de lecture

Un exemple simple va clarifier les choses :

- Exécutez le fichier **Exercice_11.swf** :

La lecture se fait en boucle, le bouton rouge sert à arrêter la lecture, le vert à la relancer, le bleu relance depuis le début, le orange bloque la tête de lecture sur la dernière image.

- Ouvrez le fichier **Exercice_11.fla** :

Les boutons Stop et Lecture sont déjà « programmés »

- Sélectionnez l'occurrence du bouton *boutonDebut* puis nommez-la *btDebut_btn*
- Sélectionnez l'occurrence du bouton *boutonFin* puis nommez-la *btFin_btn*
- Placez le script ci-dessous sur l'image clé 1 du calque Actions, à la suite du code permettant de gérer les boutons btArret_btn et btLecture_btn :

```
9   Object(root).btDebut_btn.addEventListener(MouseEvent.CLICK, allerDebut);
10  function allerDebut (evt:MouseEvent) {
11      Object(root).gotoAndPlay(1);
12  }
13  Object(root).btFin_btn.addEventListener(MouseEvent.CLICK, arreterFin);
14  function arreterFin (evt:MouseEvent) {
15      Object(root).gotoAndStop(40);
16  }
17
```

Contrôler la tête de lecture d'une animation AS 3.0

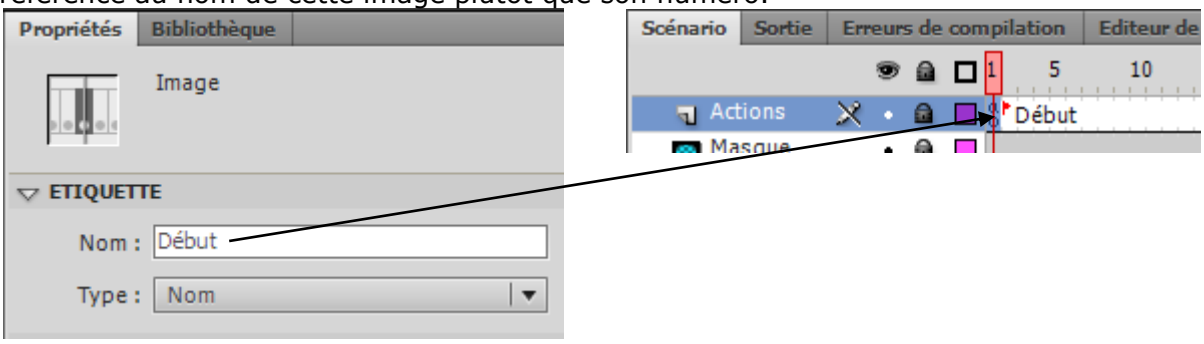
- Enregistrez votre animation, puis testez-la …

Le bouton rouge arrête la lecture, le vert la reprend, le bleu recommence tout et le jaune arrête sur la dernière image. Le programmeur progresse !

Remarque : L'ordre des instructions n'a pas d'importance, on ne peut cliquer sur 2 objets en même temps. Vous auriez pu saisir le code sous la forme suivante :

```
1  Object(root).btArret_btn.addEventListener(MouseEvent.CLICK, arretLecture);
2  Object(root).btLecture_btn.addEventListener(MouseEvent.CLICK,
   repriseLecture);
3  Object(root).btDebut_btn.addEventListener(MouseEvent.CLICK, allerDebut);
4  Object(root).btFin_btn.addEventListener(MouseEvent.CLICK, arreterFin);
5  function arretLecture(evt:MouseEvent) {
6      Object(root).stop();
7  }
8  function repriseLecture(evt:MouseEvent) {
9      Object(root).play();
10 }
11 function allerDebut (evt:MouseEvent) {
12     Object(root).gotoAndPlay(1);
13 }
14 function arreterFin (evt:MouseEvent) {
15     Object(root).gotoAndStop(40);
16 }
17
```

Vous avez la possibilité de nommer une image (Sélectionner une image clé du scénario et donner un nom via la zone de saisie à gauche de la palette Propriétés), il vous est donc possible de faire référence au nom de cette image plutôt que son numéro.



Cela pourrait ainsi donner:

```
11 function allerDebut (evt:MouseEvent) {
12     Object(root).gotoAndPlay("Début");
13 }
```

4 Déplacement vers l'image suivante : nextFrame

Cette commande s'utilise très simplement. Comparable à la fonction `gotoAndStop()`, elle permet de déplacer la tête de lecture, mais uniquement sur celle qui suit l'image actuelle.
`nextFrame()` revient à écrire `gotoAndStop(6)` lorsque la tête de lecture se trouve sur l'image clé 5.
On peut bien sûr, passer en lecture « image par image » avec cette instruction.

5 Déplacement vers l'image précédente : prevFrame()

Cette commande s'utilise très simplement, elle aussi. Comparable à la fonction `nextFrame()`, elle permet de déplacer la tête de lecture, mais uniquement sur celle qui précède l'image actuelle.
`prevFrame()` revient à écrire `gotoAndStop(4)` lorsque la tête de lecture se trouve sur l'image clé 5.

Contrôler la tête de lecture d'une animation AS 3.0

On peut bien sûr, passer en lecture « image par image en arrière» avec cette instruction.

Exemple commun avec la commande précédente :

- Ouvrez le fichier **Exercice_11_bis.fla** : les boutons Stop et Lecture sont déjà « programmés »
- Sélectionnez l'occurrence du bouton *boutonAvant* puis nommez-la *btAvant_btn*
- Sélectionnez l'occurrence du bouton *boutonArriere* puis nommez-la *btArriere_btn*
- Placez le script ci-dessous sur l'image clé 1 du calque Actions, à la suite du code permettant de gérer les boutons btArret et btLecture :

```
9   Object(root).btAvant_btn.addEventListener(MouseEvent.CLICK,
    allerAvant);
10  function allerAvant (evt:MouseEvent) {
11      Object(root).nextFrame();
12  }
13  Object(root).btArriere_btn.addEventListener(MouseEvent.CLICK,
    allerArriere);
14  function allerArriere (evt:MouseEvent) {
15      Object(root).prevFrame();
16  }
17
```

- Enregistrez votre animation, puis testez-la.

Le bouton rouge arrête la lecture, le vert la reprend, le bleu lit recule d'une image à chaque clic, alors que l' orange avance d'une image pour chaque clic. Le programmeur se joue de la tête de lecture !

Remarque : Faites attention de ne pas saisir `previousFrame()` au lieu de `prevFrame()`.

Ces deux fonctions reviennent à écrire également les scripts ci-dessous :

```
function allerAvant (evt:MouseEvent) {
   gotoAndStop(Object(root).currentframe+1);
}
function allerArriere (evt:MouseEvent) {
    gotoAndStop(Object(root).currentframe-1);
}
```

car la propriété *currentframe* de l'objet Object(root), contient le numéro de l'image placée sous la tête de lecture.

6 Contrôler la tête de lecture d'une occurrence

Nous l'avons évoqué dans la partie précédente, vous pouvez contrôler la tête de lecture du scénario d'un clip. L'intérêt de cette technique est multiple mais nous ne les développerons pas ici. Retenez simplement qu'un clip peut être comparé à une "réserve" d'images ou à une mini animation.

- Ouvrez le fichier **Exercice_11_ter.fla**.

Les boutons Stop et Lecture sont déjà « programmés », mais le scénario ne comporte plus qu'une seule image.

- Placez le script ci-dessous sur l'image clé 1 du calque Actions, à la suite du code permettant de gérer les boutons btArret_btn et btLecture_btn :

```
9   Object(root).btAvant_btn.addEventListener(MouseEvent.CLICK, allerAvant);
10  function allerAvant (evt:MouseEvent) {
11      Object(root).nextFrame();
12  }
13  Object(root).btArriere_btn.addEventListener(MouseEvent.CLICK,
    allerArriere);
14  function allerArriere (evt:MouseEvent) {
15      Object(root).prevFrame();
16  }
17
```

- Enregistrez votre animation, puis testez-la.

Contrôler la tête de lecture d'une animation AS 3.0

Rien ne va plus, le texte continue de défiler. Le programmeur s'arrache les cheveux !

- Reprenez votre script de façon à faire précéder chaque instruction de tête de lecture du nom de l'occurrence du texte suivie d'un point, comme cela :

```
9   Object(root).btAvant_btn.addEventListener(MouseEvent.CLICK, allerAvant);
10  function allerAvant (evt:MouseEvent) {
11      Object(root).defileTexte_mc.nextFrame();
12  }
13  Object(root).btArriere_btn.addEventListener(MouseEvent.CLICK,
    allerArriere);
14  function allerArriere (evt:MouseEvent) {
15      Object(root).defileTexte_mc.prevFrame();
16  }
17
```

- Enregistrez votre animation, puis testez-la.

Le bouton rouge arrête la lecture, le vert la reprend, le bleu fait reculer d'une image à chaque clic, alors que l'orange avance d'une image pour chaque clic, et tout ça avec une animation ne comportant qu'une seule image. Le programmeur danse de joie !

7 Appeler une page HTML depuis une animation

Ou plutôt, atteindre une URL avec `navigateToURL()` .
Cette fonction s'avère indispensable dès lors que vous avez besoin d'atteindre une page HTML. Sa mise en application est très simple :

```
31  var adressURL:URLRequest = new URLRequest("http://flash.zoliv.fr")
32  navigateToURL(adresseURL);
33
```

Le paramètre de cette commande n'est rien d'autre que l'adresse de la page demandée (URL), transmis sous la forme d'une variable de type URLRequest. L'URL est placée dans la variable, entourée par des guillemets (texte et non une variable).
Si vous souhaitez indiquer une frame pour le chargement de la page, ajoutez le nom de la frame comme second paramètre de cette commande.
Pour les novices, si vous ne vous sentez pas l'âme d'un programmeur, vous pourriez utiliser cette commande pour ouvrir le logiciel de messagerie afin que le visiteur de votre site vous laisse un message.

```
Var adresseURL :URLRequest = URLRequest("mailto:monadresse@monfournisseur.com")
```

De quoi s'amuser un petit moment !
On peut aussi chercher à juste cliquer sur du texte …cela se passe dans les propriétés du texte, dans la section Options, le champ Lien tandis que Cible permet de préciser la fenêtre dans laquelle doit s'afficher la page html. L'hypertexte n'est pas mort !