15

La gestion de sites Full Flash

Introduction

Par gestion de sites Full Flash, nous entendons le recours exclusif à la technologie Flash pour mettre en forme un contenu web. Le contenu affiché est étendu à la surface de la fenêtre du navigateur la plus large possible, voire, sur toute la surface de l'écran.

Naturellement, même dans un site tout en Flash, une première page HTML sera nécessaire pour le contenir (généralement "index.html"). C'est, entre autres, dans cette page – à l'aide de paramètres HTML – que nous spécifions aussi la manière dont le contenu Flash peut être déployé, avec un affichage élastique ou en mode plein écran. Nous complétons ces paramétrages à l'aide d'instructions codées naturellement en ActionScript.

Réaliser un site tout en Flash peut rapidement isoler l'utilisateur dans un dispositif hermétique. Nous vous recommandons de répondre aux exigences minimales des règles d'ergonomie et d'accessibilité lorsque vous choisissez ce type de présentation. Dans ce chapitre, nous proposons une option pour basculer l'affichage en mode plein écran, à un mode normal. Ceci afin de permettre à l'utilisateur de revenir librement à d'autres tâches sur son poste de travail sans se sentir obligé de subir un mode de navigation imposé.

Nous allons voir aussi comment organiser les contenus dans Flash, afin de rendre le site structurellement élastique avec toutes les dimensions d'écran, mais surtout, comment ne pas distordre les contenus distribués lorsqu'ils sont redimensionnés.

Il est important enfin de pouvoir intégrer tout type de contenu dans un site réalisé exclusivement en Flash. Nous parcourons également, en fin de chapitre, la gestion du PDF à travers Flash ainsi que l'importation de la typographie y compris à travers des champs de texte dynamiques.

À l'issue de ce chapitre, vous serez en mesure de créer des interfaces étendues qui valorisent spatialement les contenus.

Basculer en mode plein écran et restaurer l'affichage standard

Qu'y a-t-il de plus valorisant que de voir son travail projeté en plein écran ? Lorsque l'affichage en mode plein écran est employé à bon escient, cette option peut réellement offrir une nouvelle dimension pour organiser les contenus. La surface alors disponible pour les déployer, pour un écran de résolution standard de 1024×768 pixels, passe radicalement de 960×550 pixels, en moyenne, à 1024×768 pixels. Les images, les vidéos, la 3D,

prennent une toute autre dimension. Mais, naturellement, étendre la surface des contenus suppose quelques précautions :

- La bande passante de l'utilisateur, d'abord, elle, ne change pas. Il convient donc de ne pas profiter de cette surface supplémentaire pour augmenter le poids des contenus à charger.
- L'utilisateur qui navigue ponctuellement dans votre création a en outre besoin de pouvoir quitter le mode plein écran facilement pour revenir à une autre tâche. Nous devons donc organiser la gestion de l'affichage de sorte que l'ergonomie reste viable en toute circonstance, avec un bouton de restauration de l'affichage en mode normal, par exemple.

Dans cette section, nous étudions le basculement de l'affichage en mode plein écran, ainsi que la restauration de l'affichage en mode normal.



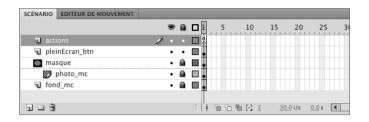
Exemples > ch15_siteFullFlash_1.fla

Dans le document "ch15_siteFullFlash_1.fla", sur la scène principale, nous pouvons voir le MovieClip fond_mc, une image isolée dans un autre MovieClip, puis un bouton et un masque (voir Figures 15.1 et 15.2). À droite de la scène, le symbole pleinEcran_btn contient deux boutons répartis sur deux images distinctes. C'est un bouton à "bascule". Lorsque l'affichage est normal, la première image est affichée. Mais dès que l'affichage bascule en mode plein écran, c'est la seconde image qui apparaît.

Figure 15.1Aperçu
de la scène
principale.



Figure 15.2 Scénario de la scène principale.



Dans la fenêtre de scénario, le calque actions affiche le code suivant :

Le code de la page HTML qui accompagne ce document ("ch15_siteFullFlash_1.html"), intègre le Flash en utilisant les paramètres suivants :

Dans notre exemple, le mécanisme de l'affichage plein écran s'exécute sous contrôle du navigateur. Le document Flash n'est interprété que par le lecteur Flash. Afin d'autoriser le navigateur à basculer l'affichage du Flash en mode plein écran, nous devons donc ajouter un paramètre dans la page HTML qui accueille le Flash.

Pour autoriser l'affichage en mode plein écran, dans la page HTML, au niveau des balises qui gèrent l'affichage du document Flash, vous devez ajouter le paramètre AllowFull-Screen et le définir sur true, y compris dans la balise embed :

```
<param name="allowFullScreen" value="true" />
```

Pour ce faire, vous pouvez l'ajouter manuellement à votre mise en forme, en saisissant directement dans le code. Vous pouvez aussi générer automatiquement une page HTML simple, avec ce paramètre, depuis Flash.

- 1. Pour publier une animation au format SWF avec une page HTML qui autorise l'affichage en mode plein écran, dans Flash, affichez les paramètres de publication, en faisant Fichier > Paramètres de publication.
- 2. Dans l'onglet HTML, dans le menu Modèle, sélectionnez l'option Flash seulement avec Autorisation du Plein écran (voir Figure 15.3).

Figure 15.3
Paramètres de publication
HTML.



3. Puis, cliquez sur le bouton Publier, en bas de cette fenêtre, pour générer le document HTML et SWF.

Attention, le code HTML généré par Flash utilise trois portes pour inclure un objet SWF: la balise object, la balise embed et une fonction JavaScript: AC_FL_RunContent. Les navigateurs modernes se servent de ce JavaScript pour implémenter le Flash dans la page. Il faut donc s'assurer que les paramètres adéquats sont également bien présents dans cette partie du code:

```
'allowFullScreen', 'true',
```

Pour tester l'affichage en plein écran, il faudra maintenant exécuter le document SWF à partir de la page HTML, dans le navigateur, et non simplement à partir du fichier SWF luimême.

Une fois que la page HTML dispose du paramètre AllowFullScreen, les actions du document Flash qui définissent l'affichage, peuvent être ajoutées.

La programmation d'un système d'affichage en mode plein écran, dans Flash, est astreinte à l'événement MouseEvent. Les autres événements n'autorisent pas ce type d'affichage pour

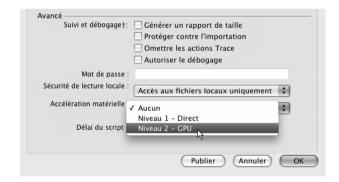
éviter naturellement les utilisations malveillantes de cette fonctionnalité. Seule une participation intentionnelle de l'utilisateur peut activer ce mode d'affichage.



Désactivation des contrôles du clavier en plein écran. Notez que l'affichage en mode plein écran désactive les contrôles du clavier. Seul une exportation au format AIR préserve ces options, mais ce format n'est réservé que pour le développement d'applications de bureau, hors fenêtre du navigateur. Nous ne l'utilisons pas dans le cadre de la réalisation d'un site web. Pour en savoir plus sur la technologie AIR, consultez le site http://www.adobe.com/fr/products/air/.

Activer l'accélération matérielle pour les contenus riches. Par défaut, le contenu Flash est exécuté avec le processeur. Pour améliorer les performances d'affichage des contenus riches (vidéos, 3D, calculs graphiques permanents) affichés en plein écran par exemple, il est possible d'activer l'accélération matérielle du poste utilisateur. Pour cela, utilisez le paramètre wmode avec la valeur GPU, directement dans le code de la page HTML qui contient le Flash. Si la carte graphique du poste client est assez puissante, elle prendra en charge l'affichage du contenu Flash. Attention, ce mode peut modifier légèrement l'aspect des contenus. Il doit, de préférence, n'être activé que pour un seul document SWF à la fois, afin de garantir la stabilité de l'affichage dans la fenêtre du navigateur. Pour activer ce mode depuis Flash, dans les paramètres de publication, dans la partie avancée de l'onglet Flash, sélectionnez l'option Niveau 2-GPU du menu Accélération matérielle (voir Figure 15.4).

Figure 15.4Activer l'accélération matérielle.



Dans le document Flash, le programme démarre sur l'initialisation :

```
//----- initialisation
pleinEcran_btn.gotoAndStop(1);
pleinEcran_btn.buttonMode=true;
```

Nous indiquons dans un premier temps de caler le bouton à bascule, sur la première image avec la méthode gotoAndStop(). Afin de laisser également apparaître la main au survol du MovieClip, nous passons la propriété buttonMode sur true.

Plus bas, un écouteur affecte une action au bouton sur un événement souris (Mouse-Event.CLICK). Cette action active l'affichage du mode plein écran :

```
//----- actions
pleinEcran_btn.addEventListener(MouseEvent.CLICK, affichagePleinEcran);
function affichagePleinEcran(evt:MouseEvent) {
```

```
if (stage.displayState==StageDisplayState.NORMAL) {
    stage.displayState=StageDisplayState.FULL_SCREEN;
    pleinEcran_btn.gotoAndStop(2);
} else {
    stage.displayState=StageDisplayState.NORMAL;
    pleinEcran_btn.gotoAndStop(1);
}
```

Nous pouvons basculer l'ensemble du document (stage) ou un conteneur. Il suffit de spécifier le nom de l'objet à projeter et de lui appliquer la propriété displayState qui désigne son état en plein écran d'affichage. Dans notre exemple, toute la scène est basculée en mode plein écran. C'est donc l'objet stage qui reçoit la propriété displayState. Nous définissons, pour cet objet, le type d'affichage à l'aide de la valeur stageDisplay-State.FULL SCREEN:

```
stage.displayState=StageDisplayState.FULL_SCREEN;
```

Nous avons ajouté une condition qui vérifie l'état d'affichage. Si l'affichage est normal, nous basculons en plein écran. Si, inversement, l'état d'affichage est déjà en plein écran, nous spécifions l'action inverse, de restaurer l'affichage :

```
stage.displayState=StageDisplayState.NORMAL;
```

Pour basculer d'une icône à une autre, nous ajoutons aussi une action gotoAndStop() qui cible l'image correspondant à l'action disponible pour chaque mode d'affichage. Cette condition permet d'appliquer une action à bascule sur un même et unique bouton. Mais vous pouvez bien entendu séparer les instructions pour les isoler sur des objets distincts.

En projetant la page HTML dans le navigateur, et en activant le bouton pleinEcran_btn, le site apparaît en plein écran. En cliquant à nouveau, il revient à un affichage normal (voir Figure 15.5).

Figure 15.5Affichage en plein écran.





Quitter le mode Plein écran avec Echap. Indépendamment des contrôles de bouton développés manuellement, Flash active, par défaut, une option de restauration de l'affichage si vous appuyez sur la touche Echap du clavier. Un message apparaît pour prévenir de cette option dès le passage en mode plein écran. Il n'y a pas de contrôle pour modifier ce message afin de garantir à tout utilisateur une sortie possible, quel que soit le type de développement ajouté. Mais nous pouvons détecter l'action sur le bouton esc... pour repositionner notre bouton bascule à son état normal :

```
stage.addEventListener(FullScreenEvent.FULL_SCREEN, ecoute);
function ecoute(evt:FullScreenEvent){
   if (! event.fullScreen) {
      pleinEcran_btn.gotoAndStop(1);
   }
}
```

À retenir

- Le mode d'affichage en plein écran se définit par la propriété stageDisplayState.
- Pour activer l'affichage en mode plein écran, la page HTML qui contient le Flash doit également contenir le paramètre AllowFullScreen et le passer sur true.
- Les actions de clavier sont inactives en mode plein écran.
- Le mode plein écran n'est disponible qu'à travers une action souris.

Interface élastique flottante

Le principe d'une interface élastique est d'appliquer, pendant le redimensionnement de la fenêtre du navigateur, les proportions de cette fenêtre au document Flash, dans le but de remplir systématiquement la surface de la fenêtre quelles qu'en soient les dimensions.

Cela se construit initialement de manière assez simple puisqu'il suffit de renseigner une largeur et une hauteur de 100 % dans les propriétés de dimensions du Flash du document HTML. Mais si l'on ne veut pas que les contenus soient déformés, et qu'ils occupent, malgré tout, l'intégralité de la surface de la fenêtre, seules des instructions en ActionScript permettent de redistribuer les contenus en contrôlant leurs dimensions, leurs proportions et leurs positions en X et Y.

Dans cette section, nous allons voir comment placer des MovieClip dans la scène de sorte qu'elle s'étende, quelle que soit la taille de la fenêtre, mais sans jamais distordre les contenus ni laisser de vide dans le document.



Exemples > ch15_siteFullFlash_2.fla

Dans le document "ch15_siteFullFlash_2.fla", sur la scène, nous distinguons à l'arrière-plan une image de 800×600 pixels placée dans un MovieClip contenu_mc (voir Figure 15.6). Au-dessous, à droite, apparaît un MovieClip texteDroite_mc, qui contient le titre du livre. À gauche, un autre MovieClip affiche le logo de l'éditeur. Le filet est intégré dans le clip du

bandeau bleu, calé, lui, en bas de la fenêtre. Tout en haut du document, apparaît aussi un symbole nommé carte_mc qui représente le département du Vaucluse.

Figure 15.6 Aperçu du document.



Figure 15.7 Scénario de la scène principale.

SCÉNARIO EDITEUR DE MOUVEMENT								
		9 6		5	10	15 20	25	30
¬ actions	2		□ \$					
□ carte_mc			· 🔲 🌡					
√ logo_mc			· 🔳 👢					
texteDroite_mc text			· 🔳 🕽					
■ bandeau_mc			· 🔲 🌡					
¬ contenu_mc			· 🔳 🌡					
3 D 3				66	% [:] 1	30.0 i/s	0.0 s	1

Dans la fenêtre Actions (voir Figure 15.7), nous pouvons lire le code suivant :

```
//
bandeau_mc.x=stage.stageWidth/2;
bandeau_mc.y=stage.stageHeight-(bandeau_mc.height/2);
bandeau_mc.width=stage.stageWidth;
//
contenu_mc.x=0;
contenu_mc.y=0;
contenu_mc.width=stage.stageWidth;
contenu_mc.height=stage.stageHeight;
if (contenu_mc.scaleX<=contenu_mc.scaleY) {
   contenu_mc.scaleX=contenu_mc.scaleY;
} else {
   contenu_mc.scaleY=contenu_mc.scaleX;
}
modifierAffichage();</pre>
```

La scène, en ActionScript 3, possède des propriétés de largeur (stageWidth) et de hauteur (stageHeight), que nous avons déjà rencontrées dans les chapitres précédents. Grâce à ces propriétés, nous connaissons à tout moment les dimensions du document. Nous pouvons donc récupérer ces valeurs pour redéfinir, dynamiquement, l'emplacement des objets par rapport à la taille du document.

Nous pouvons, par exemple, jouer sur les positions X et Y, pour caler systématiquement un logo en bas et à gauche. Nous pouvons, aussi, dans le cas où certains éléments doivent être redimensionnés, modifier leurs propriétés width et height en proportions des largeur et hauteur de la scène.

Dans cet exemple, nous plaçons les objets tantôt selon la largeur du document, tantôt selon sa hauteur. Pour deux d'entre eux, l'image de fond ainsi que le bandeau du bas, nous redéfinissons leurs dimensions en proportion des dimensions de la scène.

Dans la première partie du code, nous définissons le comportement de l'affichage de la scène par défaut :

```
//---- initialisation
stage.scaleMode=StageScaleMode.NO_SCALE;
```

La propriété scaleMode permet d'indiquer si les contenus sont déformés ou non lorsque la taille de la fenêtre du document est modifiée. La valeur stageScaleMode.NO_SCALE indique qu'aucune déformation n'est appliquée aux contenus.



Modes d'affichage de la scène. Il existe quatre modes d'affichage de la scène courante : NO_SCALE, EXACT_FIT, NO_BORDER et SHOW_ALL. Elles définissent le comportement des objets dans la scène, lors du redimensionnement du document Flash dans la fenêtre de navigateur. Les modifications de propriété width et height, ajoutées éventuellement en ActionScript sur des objets demeurent prioritaires sur ces options-ci.

- NO_SCALE empêche les modifications d'échelle des contenus même si les dimensions de la fenêtre sont modifiées.
- EXACT_FIT autorise la distorsion des contenus selon l'exacte proportion des dimensions de la fenêtre.

- NO_BORDER préserve les proportions de la scène quelle que soit la taille de la fenêtre. La scène est agrandie et recadrée si nécessaire. Le contenu occupe toujours intégralement la fenêtre.
- SHOW_ALL préserve également les proportions de la scène et l'agrandit si nécessaire, mais ne recadre jamais le contenu. Des marges sont ajoutées autour de la scène si les proportions de la fenêtre sont différentes de celles de la scène.

La deuxième instruction emploie la propriété align qui définit l'alignement de la scène, lorsque la fenêtre est redimensionnée :

```
stage.align=StageAlign.TOP LEFT;
```

Dans notre exemple, la valeur TOP_LEFT désigne un calage en haut et à gauche. Ainsi, si nous n'avions pas modifié dynamiquement le positionnement de tous les objets, ils resteraient à leur place actuelle et le redimensionnement de la fenêtre impliquerait l'ajout de marges à droite et en bas du document.

Propriétés d'alignement de la scène

- BOTTOM. La scène est alignée sur le bas.
- BOTTOM_LEFT. La scène est alignée sur le coin inférieur gauche.
- BOTTOM RIGHT. La scène est alignée sur le coin inférieur droit.
- LEFT. La scène est alignée sur la gauche.
- RIGHT. La scène est alignée sur la droite.
- TOP. La scène est alignée sur le haut.
- TOP LEFT. La scène est alignée sur le coin supérieur gauche.
- TOP RIGHT. La scène est alignée sur le coin supérieur droit.

Nous détectons ensuite le redimensionnement de la fenêtre en ajoutant un écouteur à la scène (stage). La propriété RESIZE permet d'exécuter la fonction durant chaque redimensionnement de fenêtre et uniquement là :

```
//----- actions
stage.addEventListener(Event.RESIZE, quandResizeActif);
function quandResizeActif(event:Event) {
   modifierAffichage();
}
```

La fonction invoquée appelle une autre fonction, autonome :

```
modifierAffichage();
```

Nous avons choisi d'isoler les actions dans une deuxième fonction afin de permettre de les appeler depuis différents emplacements et selon différents types d'événements. En l'occurrence, nous souhaitons exécuter les instructions pendant le redimensionnement, mais aussi dès l'ouverture du document. Si nous ne redéfinissions pas le positionnement dès l'ouverture de l'animation, nous risquerions en effet d'observer un saut ou un décalage entre la position courante et celle qui est redéfinie au premier redimensionnement.

La fonction appelée redéfinit les propriétés x, y, width et height des différents contenus selon les largeurs et hauteur de la scène :

```
function modifierAffichage() {
   //
  logo mc.x=20;
  logo mc.y=stage.stageHeight-35;
  //
  texteDroite mc.x=stage.stageWidth-20;
  texteDroite mc.y=stage.stageHeight-35;
   11
   bandeau mc.x=stage.stageWidth/2;
  bandeau mc.y=stage.stageHeight-(bandeau mc.height/2);
  bandeau mc.width=stage.stageWidth;
  contenu mc.x=0;
  contenu mc.y=0;
  contenu mc.width=stage.stageWidth;
  contenu mc.height=stage.stageHeight;
  if (contenu mc.scaleX<=contenu mc.scaleY) {</pre>
      contenu mc.scaleX=contenu mc.scaleY;
   } else {
      contenu mc.scaleY=contenu mc.scaleX;
  }
}
```

Les premières lignes récupèrent simplement les valeurs stage.stageWidth et stage.stage-Height. Des marges sont toutefois ajoutées ou soustraites en fonction de la position réellement souhaitée pour chaque objet. Par exemple, le symbole logo_mc est placé systématiquement à 20 pixels du bord gauche de la fenêtre, mais toujours calé en bas à 35 pixels du bord (stage.stageHeight-35 équivaut à dire : la hauteur moins 35 pixels).

Lorsque vous positionnez les objets, pensez à tenir compte de leur point d'alignement (centre) et de leur hauteur ou largeur. Ceci est déterminent pour caler parfaitement les éléments dans la scène.

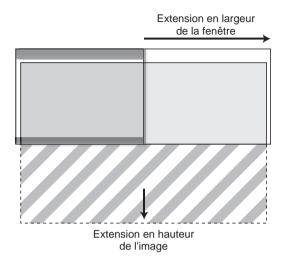
Le MovieClip bandeau_mc est positionné en Y selon la même hauteur de fenêtre. Mais il est également redimensionné en largeur (width) selon la largeur effective de la fenêtre. Le symbole occupe donc toute la largeur de la scène quelle qu'en soit la taille. Nous avons bien entendu pris soin d'y placer des graphismes qui supportent cette distorsion.

La dernière partie de la fonction affecte également les positions X et Y, ainsi que les dimensions width et height de l'objet contenu_mc. Mais, nous ajoutons ici une condition :

```
contenu_mc.x=0;
contenu_mc.y=0;
contenu_mc.width=stage.stageWidth;
contenu_mc.height=stage.stageHeight;
if (contenu_mc.scaleX<=contenu_mc.scaleY) {
   contenu_mc.scaleX=contenu_mc.scaleY;
} else {
   contenu_mc.scaleY=contenu_mc.scaleX;
}</pre>
```

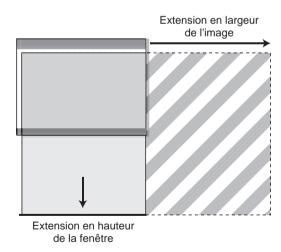
Lorsque la fenêtre est étirée en largeur, pour remplir la surface de la fenêtre, nous devons étirer le contenu également en largeur. Mais, afin de préserver ses proportions, nous devons l'étirer aussi dans le sens de la hauteur, hors champ (voir Figure 15.8).

Figure 15.8 Étirement du contenu à la verticale.



Inversement, lorsque la fenêtre est étirée en hauteur, nous devons agrandir le contenu en hauteur et, indirectement, en largeur, hors champ (voir Figure 15.9).

Figure 15.9 Étirement du contenu à l'horizontal.



Il en résulte que l'image remplit la surface de la fenêtre en toutes circonstances.

Mais, nous ne pouvons à la fois dire que la largeur du contenu correspond à la largeur de la fenêtre et que la hauteur de ce même contenu équivaut également la hauteur de la fenêtre. Car il suffit que la fenêtre n'affiche pas les mêmes proportions que celles du contenu pour que les instructions entrent en conflit. Pour résoudre ce dilemme, nous devons spécifier une action qui active l'un ou l'autre des redimensionnements, en fonction de la plus grande

longueur observée. Si c'est la largeur qui est plus grande que la hauteur, alors, nous étirons l'image en hauteur. Si, en revanche, c'est la hauteur de la fenêtre qui est plus importante, alors, nous étirons l'image dans sa largeur. Comme, dans les deux cas, l'image conserve ses proportions, elle remplira toujours la surface de l'écran.

Nous précisons alors, dans notre fonction, la condition suivante :

```
if (contenu_mc.scaleX<=contenu_mc.scaleY) {
      contenu_mc.scaleX=contenu_mc.scaleY;
   } else {
      contenu_mc.scaleY=contenu_mc.scaleX;
}</pre>
```

Si l'échelle de redimensionnement du contenu en largeur (scaleX) est inférieure à son redimensionnement en hauteur (scaleY) (donc, si la fenêtre est plus haute que large), alors, l'échelle en largeur (scaleX) prend pour valeur l'échelle de la hauteur (scaleY):

```
if (contenu_mc.scaleX<=contenu_mc.scaleY) {
      contenu_mc.scaleX=contenu_mc.scaleY;
}</pre>
```

Inversement, si c'est la largeur qui est plus grande que la hauteur, dans ce cas, nous appliquons l'échelle de la hauteur (scaleY) sur l'échelle de la largeur (scaleX) :

```
else {
     contenu_mc.scaleY=contenu_mc.scaleX;
}
```

Cette technique permet de redimensionner la fenêtre dans des proportions libres, différentes de celles de la scène, et d'élargir quand même le contenu, mais sans jamais y appliquer de distorsion. Seule une transformation homothétique est appliquée.

Pour assurer une remise à l'échelle de qualité correcte, pensez également à appliquer, aux contenus, un lissage. Pour en savoir plus sur cette technique, reportez-vous au chapitre consacré aux API d'affichage et de filtres. Dans ce document, nous avons simplement appliqué un lissage depuis la fenêtre Bibliothèque.



Principe de flottement façon styles CSS. En déclinant le principe de redimensionnement, ici évoqué, et celui du repositionnement, développé dans la section des menus déroulants, vous pouvez aussi concevoir une interface dans laquelle les objets flottent de manière relative, les uns par rapport aux autres, et se redimensionnent selon la taille effective de la fenêtre du navigateur, à la manière d'une page HTML, par exemple.

À retenir

- Pour placer et redimensionner des objets en fonction de la taille de la fenêtre, nous utilisons les propriétés stageWidth et stageHeight de la scène.
- Afin d'appliquer une mise à l'échelle homothétique, nous ajoutons une condition qui détermine le redimensionnement en fonction de la largeur la plus grande. Cette largeur est ensuite utilisée pour définir le cœfficient de redimensionnement.

Gérer le PDF

Une interface tout en Flash implique la nécessité d'y importer différents types de médias, afin, tout simplement, de ne pas avoir à sortir de cet espace pour y accéder. Typiquement, cela pose problème lorsqu'un lien, dans Flash, cible un document PDF. Dans ce contexte, le format PDF, qui requiert l'ouverture d'Acrobat, oblige l'utilisateur à sortir du site tout en Flash pour pouvoir lire un contenu.

Dans cette section, bien que cela soit structurellement impossible, nous abordons les contournements qui permettent d'intégrer malgré tout un document PDF dans Flash, afin de permettre à l'utilisateur de rester dans son univers développé tout en Flash.



Différence structurelle entre Flash et Acrobat. Flash compile des données à la publication du document. Acrobat, à l'inverse, ne gère qu'un amoncèlement de scripts assemblés dans un certain ordre, sans les compiler. Acrobat agit un peu à la manière d'une page HTML qui gère des fonctions JavaScript et de médias zippés. La structure des documents étant très différente, l'implémentation du format non compilé dans le format compilé est peu concevable. L'inverse, en revanche, se fait facilement. Vous pouvez parfaitement placer un contenu SWF dans un document Acrobat.

Il n'est pas possible d'afficher dynamiquement un PDF dans l'interface de Flash. Mais, il est possible d'importer un PDF physiquement, dans la scène, avant publication. Il est également possible de générer dynamiquement des fichiers au format PDF à partir d'un document Flash. Vous pouvez aussi appeler un fichier PDF à travers un hyperlien en utilisant la méthode navigateToURL, abordée au Chapitre 14. Quelques services en ligne proposent aussi de vous fournir dans un format compatible Flash, un PDF recompilé.

Pour importer physiquement un PDF dans Flash, vous devez d'abord le convertir dans un autre format *via* InDesign ou Illustrator, par exemple, les formats compatibles avec Flash sont l'EPS, le format AI et le XFL. Naturellement, les formats d'images sont également compatibles, mais alourdissent considérablement le poids des contenus et la gestion des documents.

- 1. Depuis InDesign, faites Fichier > Exporter (Ctrl+E pour Windows ou Cmd+E pour Macintosh).
- 2. Puis, dans la fenêtre d'exportation, dans le menu Format, choisissez "Adobe Flash CS4 prof (XFL)". Validez.
- 3. En validant, une boîte de réglage apparaît (voir Figure 15.10).
- 4. Spécifiez un texte InDesign en texte Flash, afin de préserver l'édition du document. Puis, validez.
- 5. Depuis Flash, faites ensuite Fichier > Ouvrir.
- 6. Sélectionnez le document XFL et confirmez l'importation.

Le document est importé dans la scène et vous pouvez l'animer. Le procédé est similaire avec Illustrator.

Vous pouvez aussi convertir un PDF en livre à feuillets depuis InDesign. Reportez-vous également au Chapitre 9 pour en savoir plus à ce sujet.

Figure 15.10 Scénario de la scène principale.





Convertir un PDF en SWF avec FlashPaper. Il est possible de générer un document Acrobat PDF en fichier Flash SWF, à la volée. L'utilitaire FlashPaper, développé par la société Macromedia, et compatible uniquement avec Windows, offre cette possibilité. Une version d'évaluation de ce programme est disponible à l'adresse suivante : http://www.adobe.com/cfusion/tdrc/index.cfm?product=flashpaper.

Création dynamique de fichiers PDF. Pour générer dynamiquement un PDF, nous pouvons utiliser la classe byteArray. Reportez-vous au Chapitre 5 du livre de Thibault Imbert, éditions Pearson, *Pratique d'ActionScript 3*, pour en savoir plus à ce sujet.

À retenir

 Il est possible d'importer un document PDF au sein d'une animation Flash. Pour cela, nous devons au préalable le convertir en dur depuis une application graphique ou dynamiquement.

Gérer la typographie

La gestion de la typographie, dans une page web, soulève toujours de nombreuses interrogations. Usuellement, une page web dépend des polices présentes sur le système de l'utilisateur pour afficher un texte. Dans une page HTML, nous spécifions alors la police à employer. Si celle-ci est absente de la configuration de l'utilisateur, c'est la police activée par défaut, par le navigateur, qui prend le relais (généralement Times).

Dans Flash, c'est différent. Flash propose trois modes de traitement pour le texte : les textes statiques, les textes dynamiques et les textes de saisie. Nous ne comptons pas les textes vectorisés sous forme de graphisme ni les textes véhiculés sous la forme d'images.



Vectoriser un texte dans Flash. Pour vectoriser un texte dans Flash, faites deux fois de suite Ctrl+B (Windows) ou Cmd+B (Mac). Cette commande casse l'enveloppe courante, qu'elle soit un bloc de texte ou un symbole. Pour le texte, la première action sépare les lettres, la seconde les vectorise. Petit rappel mnémotechnique : pensez à B comme *Break* qui signifie casser en anglais. Une fois le texte vectorisé, pensez également à le regrouper afin d'éviter tout risque de morcellement si vous

devez ensuite le manipuler. Pour le regrouper, faites Ctrl+G (Windows) ou Cmd+G (Mac). Attention, un texte vectorisé n'est plus éditable. Mais la forme graphique devient modifiable avec les outils de dessin (Plume, pot de peinture, outils de sélection).

Les textes statiques

Les champs de textes statiques intègrent la forme de contour vectorielle des caractères, à la publication du document. Les caractères sont donc pris en charge nativement.



Affichage des polices dans Flash. Attention, toutes les polices affichées correctement dans le document Flash ne sont pas nécessairement intégrées à la publication. Pour vérifier qu'une police est intégrée, vérifiez que l'option de menu Affichage > Mode Aperçu > Texte anti-aliasé, est bien cochée. Si une police s'avère non intégrable, elle apparaît alors avec un crénelage dans le document. Vous pouvez ainsi en choisir une autre depuis le menu Famille de caractère, de l'inspecteur de propriétés.

Les documents Flash peuvent utiliser des polices Type 1 PostScript®, TrueType® et bitmap (uniquement sur Macintosh) de taille maximum de 255 points. Les textes sont Unicode depuis la version du lecteur Flash 7.

Les textes dynamiques

Les textes dynamiques et de saisie, en revanche, utilisent les polices de périphérique (_sans, _serif, ou _typewriter). Ces polices sont disponibles en tête de liste dans la liste des polices de l'inspecteur de propriétés. La police _sans correspond à de l'Arial ou de l'Helvetica. La police _serif, à un Times. La police _typewriter à un Courier.

Si vous souhaitez ajouter une typographique spécifique à un texte dynamique, cela reste possible. Il suffit d'en ajouter les caractères nécessaires, sur chaque champ concerné, depuis l'inspecteur de propriétés.

- 1. Pour ajouter une police à un champ de texte dynamique, sélectionnez le champ de texte avec l'outil de sélection (flèche noire).
- 2. Dans l'Inspecteur de propriétés, cliquez sur Intégration de caractères (voir Figure 15.11).
- 3. Une boîte de dialogue s'ouvre.
- 4. Dans la liste qui apparaît, sélectionnez uniquement les caractères utiles pour optimiser autant que possible le poids du document (voir Figure 15.12). Au besoin, ajoutez manuellement les caractères manquants, dans le champ Inclure ces caractères.

Lorsque vous ajoutez des caractères manuellement, relevez que chaque caractère ajouté augmente d'autant le poids du fichier. Éviter, par conséquent, de recourir à un nombre trop varié de polices de caractère dans un même document.

Lorsque vous saisissez les caractères à ajouter, pensez à ajouter aussi l'espace, les sigles monétaires et les caractères accentués ou spéciaux. Les listes de caractères par défaut étant conçues initialement pour un public anglo-saxon, il convient d'ajouter ceux qui ne figurent pas dans cet alphabet, comme les caractères é, è, ç, à, ô, î, û, ê, â, ö, ï, ü, ë, ä, 1 et espace.

Figure 15.11

Inspecteur de propriétés.



Figure 15.12 Intégration de caractères.



Dans la Figure 15.12, nous avons pris soin de ne sélectionner que les listes de caractères Minuscules, Majuscules, Chiffres et Ponctuations, puis de saisir, à la main, les caractères spéciaux manquants, directement dans le champ Inclure les caractères. La liste "Tous" intègre

absolument tous les caractères, vous augmenteriez considérablement le temps de publication et le poids de votre document, en la sélectionnant. Il est en outre préférable de restreindre le choix aux caractères utiles uniquement et d'y ajouter, si nécessaire, les caractères manquants, dans le champ Inclure les caractères, prévu à cet effet.

Attention, l'option Remplissage automatique n'enregistre que les caractères affichés dans le champ actif.



Les polices et les masques. Si vous utilisez des champs de textes dynamiques ou de saisie, avec des masques, vérifiez toujours la saisie ou l'affichage sur des postes utilisateurs différents. Selon la configuration du système, certains caractères peuvent ne pas apparaître. Dans ce cas, pensez déjà à ajouter les caractères manquants. Notez que le paramètre wmode utilisé pour un affichage transparent du document Flash (vu précédemment) peut aussi neutraliser certains caractères à la saisie (arobase), si un masque est utilisé, avec d'anciennes versions du lecteur Flash. Dans ce cas, évitez d'utiliser conjointement les masques et les champs de textes dynamiques.

Les bibliothèques partagées de symboles et de typographies

Le principe de la bibliothèque partagée est de centraliser, sous la forme d'un fichier SWF ou SWC, un ou plusieurs objets de la bibliothèque d'un document Flash. Cette technique permet d'optimiser la gestion d'objets récurrents sans avoir à les stocker à l'intérieur de chaque document SWF qui doit les utiliser. Sachant combien le poids d'une typographie peut être compromettant dans l'élaboration d'un site web, et combien son utilisation affecte, pourtant, les identités graphiques, rassembler une typographie dans une bibliothèque partagée peut nous affranchir de quelques contraintes.



Exemples > ch15_siteFullFlash_3.fla Exemples > ch15_siteFullFlash_3b.fla

Dans cette section, nous présentons comment exporter une police à partir d'un document Flash et l'intégrer dynamiquement à tout autre document SWF. Les deux fichiers d'exemple "ch15_siteFullFlash_3.fla" et "ch15_siteFullFlash_3b.fla" reprennent cette procédure.

Pour partager une typographie, dans le menu contextuel de la fenêtre de bibliothèque d'un document Flash, choisissez Nouvelle police. Une fenêtre de dialogue apparaît (voir Figures 15.13 à 15.15).

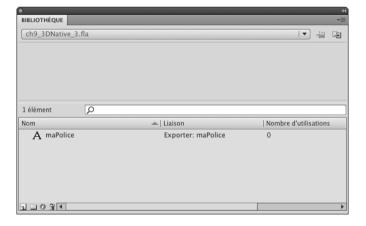
- 1. Dans le champ Nom, inscrivez le nom que vous souhaitez attribuer à votre police partagée.
- Dans le menu Police, sélectionnez la typographie de votre choix. Ajoutez éventuellement quelques options comme l'italique, le gras, si ces propriétés sont disponibles par exemple.
- 3. Dans la partie centrale, activez l'option Exporter pour ActionScript si vous souhaitez pouvoir disposer de la typographie en programmation pour des objets TextField. Et attribuez dans ce cas un nom de classe à la police.

Figure 15.13 Ajout d'une police partagée à la bibliothèque.

	Propriétés des symboles de police
Nom :	maPolice OK
Police :	AvantGarde CE Annuler
Style :	Regular
	Faux gras Texte bitmap Faux italique Taille : 94 Options de base
Liaison ——	 ✓ Exporter pour ActionScript ✓ Exporter dans l'image 1
Ider	tifiant :
	Classe : maPolice
Classe d	e base : flash.text.Font
Partage ——	☑ Exporter pour le partage à l'exécution
	URL: ch15_siteFullFlash_3.swf

4. Dans la partie inférieure, activez l'option Exporter pour le partage à l'exécution. Dans le champ URL, maintenant actif, inscrivez le nom de publication du fichier SWF de ce document Flash. Puis validez. La fenêtre de bibliothèque affiche la nouvelle police. La colonne Liaison indique que l'objet est prêt pour l'exportation (voir Figure 15.14).

Figure 15.14
Affichage de la police partagée dans la bibliothèque.



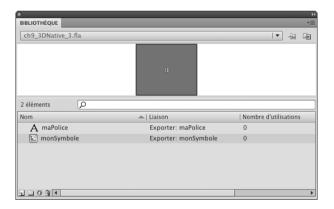
- 5. Procédez de même pour des symboles, en créant par exemple un movieClip. Puis, dans la fenêtre de bibliothèque, affichez les propriétés contextuelles du movieClip en faisant un clic-droit > Propriété, sur le symbole. La fenêtre de propriétés du symbole apparaît (voir Figure 15.15).
- 6. Activez les options d'exportation et validez.

Figure 15.15 Ajout d'un symbole partagé à la bibliothèque.



La fenêtre de bibliothèque affiche également le MovieClip et son statut d'objet pour l'exportation (voir Figure 15.16).

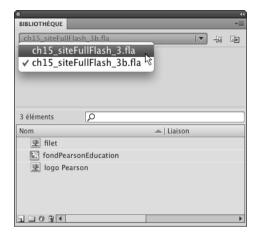
Figure 15.16Affichage du symbole partagé dans la bibliothèque.



- 7. Sans nécessairement utiliser les objets dans la scène de ce document, publiez-le au format SWF en faisant Ctrl+Entrée (Windows) ou Cmd+Entrée (Macintosh).
- 8. Puis, dans un nouveau document Flash que vous aurez préalablement enregistré, importez les éléments partagés par glisser-déposé de la bibliothèque du document exportateur vers la scène du document importateur (voir Figure 15.17).
- 9. Pour accéder à la bibliothèque d'un autre document, depuis le sommet de la fenêtre de bibliothèque du document actif, sélectionnez dans la liste déroulante, le nom du document contenant les objets à importer.

Figure 15.17

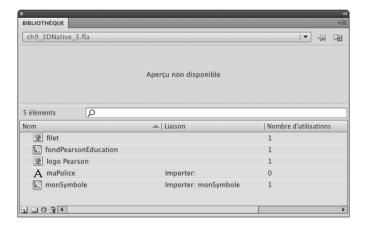
Importation dynamique des objets partagés dans un autre document..



La fenêtre de bibliothèque du nouveau document affiche le statut des objets importés une fois ceux-ci ajoutés (voir Figure 15.18).

Figure 15.18

Affichage des objets importés dans la bibliothèque du nouveau document.



- 10. Dans la scène du nouveau document, placez une occurrence du MovieClip importé et ajoutez un champ de texte en y inscrivant quelques mots. Pour appliquer la police importée au champ de texte, depuis l'inspecteur de propriétés du texte, dans le menu Famille de caractère, sélectionnez le nom que vous avez attribué à votre police, désormais disponible. Le champ de texte affiche la police sélectionnée. Dans le document "ch15_siteFullFlash_3b.fla", nous utilisons les objets partagés du document "ch15_siteFullFlash_3.fla" (voir Figure 15.19). Publiez le nouveau document au format SWF.
- 11. Puis, revenez dans le document initial pour modifier la forme intérieure du MovieClip original et la police. Pour la modifier, depuis les propriétés de la police, disponibles à partir de la fenêtre de bibliothèque, sélectionnez une autre famille de caractères et validez votre modification. Publiez à nouveau ce document.

Figure 15.19
Utilisation
des objets
partagés dans
le document..



12. Sans republier le nouveau fichier Flash qui importe les objets partagés, exécutez directement son format déjà publié en SWF. Les objets sont instantanément mis à jour.



Le nouveau moteur de texte Text Layout Framework. Adobe prévoit l'implémentation d'un nouveau moteur de gestion de texte pour les versions à venir de Flash. Il permet de gérer le flottement du texte sur plusieurs colonnes, dynamiquement, la chasse, l'interlignage, la sélection, comme dans un véritable logiciel de mise en pages. Ce moteur est actuellement disponible en version bêta. Vous pouvez le tester à l'adresse suivante : http://labs.adobe.com/technologies/textlayout/.

À retenir

Il est possible de partager des symboles et des polices de caractère pour plusieurs documents Flash, simultanément. Pour cela, nous activons l'option Exporter pour le partage à l'exécution, depuis les propriétés des objets disponibles à partir de la bibliothèque et importons cette bibliothèque dans d'autres documents Flash.

Importer Flash AS1/AS2 dans Flash AS3

Les documents codés en AS1 ou AS2 utilisent le même moteur (AVM1). Le langage ActionScript 3 est, lui, structurellement différent des précédentes versions et utilise un nouveau moteur (AVM2). Les documents Flash basés sur AS3 disposent donc d'une architecture différente des anciens documents Flash. Si nous pouvons facilement importer des contenus AS1 ou AS2 animés dans un document AS3, à partir d'un simple chargeur, nous ne pouvons pas, en revanche, y apporter de l'interactivité. Pour cela, nous devons ouvrir une connexion entre les deux formats de fichier, par nature dissociés, à l'aide de la classe LocalConnexion.

Pour comprendre comment utiliser cette classe, nous devons admettre que le fichier AS1/AS2 d'une part et AS3 d'autre part, sont hermétiques l'un envers l'autre, et que seul un objet localConnexion permet d'ouvrir une porte de chaque côté des deux animations pour transmettre des requêtes d'AS3 vers AS2.

Il devient alors facile d'invoquer une fonction AS2, depuis une instruction placée dans un document AS3, par simple identification du nom de la connexion et de la fonction, définies dans le document AS2.

Le fichier AS3 émetteur est appelé fichier d'envoi. Il contient la méthode à appeler. Le fichier d'envoi doit contenir un objet LocalConnection et un appel de la méthode send(). L'autre fichier, le fichier AS1 ou AS2, est dit de réception. Il s'agit de celui qui appelle la méthode. Le fichier de réception doit contenir un autre objet LocalConnection et un appel de la méthode connect().

Dans cette section, nous présentons un document AS3 qui contient un bouton. Ce bouton affecte les propriétés d'un symbole de type MovieClip, contenu dans un fichier AS2 importé.



Exemples > ch15_siteFullFlash_4_AS3.fla Exemples > ch15_siteFullFlash_4b_AS2.fla

Le document "ch15_siteFullFlash_4_AS3.fla" affiche une scène exportée pour Flash 10 et ActionScript 3.0. Sur cette scène figure un bouton nommé connexion_btn (voir Figure 15.20).

Figure 15.20 Aperçu du document AS3.



Le fichier "ch15_siteFullFlash_4b_AS2.fla", affiche quant à lui une scène exportée pour ActionScript 2.0 et le lecteur Flash 8. Sur cette scène, figure un MovieClip nommé symbole mc (voir Figure 15.21).

Figure 15.21 Aperçu du document AS2.



En publiant le document AS3, le fichier AS2 est d'abord importé en premier plan. En cliquant sur le bouton connexion_btn, du document AS3, le symbole du fichier AS2 subit une rotation de 20° (voir Figure 15.22).

Figure 15.22 Interaction AS3 vers AS2.



Dans la fenêtre Actions du document AS2 ("ch15_siteFullFlash_4b_AS2.fla"), nous pouvons lire le code suivant :

```
// récepteur
var recepteur:LocalConnection = new LocalConnection();
```

```
// active la connexion
recepteur.connect("maConnexionAS2");

// définition de la fonction AS2 activée depuis le document AS3
recepteur.monActionProgrammeeAS2=function () {
    // actions AS2
    symbole_mc._rotation+=20;
}
```

Dans le code du document AS3 ("ch15_siteFullFlash_4_AS3.fla"), nous lisons le programme suivant :

```
// chargeur
var chargeur:Loader = new Loader();
var chemin:URLRequest=new URLRequest("ch15_siteFullFlash_4b_AS2.swf");
chargeur.load(chemin);
addChild(chargeur);

// connexion avec SWF AS2
var emetteur:LocalConnection = new LocalConnection();
connexion_btn.addEventListener(MouseEvent.CLICK,AccesAS2);
function AccesAS2(evt:MouseEvent) {
    // nom de la connexion définie dans AS2, nom de la fonction définie dans AS2
    emetteur.send("maConnexionAS2", "monActionProgrammeeAS2");
}
```

Dans le premier document (AS2), nous commençons par ouvrir une connexion avec un objet LocalConnexion :

```
// récepteur
var recepteur:LocalConnection = new LocalConnection();
```

À la suite, nous activons cette connexion, à l'aide de la méthode connect(). En paramètre, nous spécifions le nom de cette connexion invoquée plus tard dans le fichier AS3.

```
// active la connexion
recepteur.connect("maConnexionAS2");
```

Puis, sur cet objet, nous définissons une fonction dont le nom (monActionProgrammeeAS2) sera également invoqué plus tard dans le document AS3.

```
// définition de la fonction AS2 activée depuis le document AS3
recepteur.monActionProgrammeeAS2=function () {
   // actions AS2
   symbole_mc._rotation+=20;
}
```

Pour terminer, l'instruction applique une rotation de 20° sur le clip symbole_mc placé sur la scène principale (symbole mc. rotation+=20).

Dans le document AS3 appelant, dans la fenêtre des actions, nous commençons par charger le fichier SWF codé en AS2. Puis, plus loin, nous ouvrons également une connexion en définissant un nouvel objet LocalConnexion :

```
// connexion avec SWF AS2
var emetteur:LocalConnection = new LocalConnection();
connexion_btn.addEventListener(MouseEvent.CLICK,AccesAS2);
```

```
function AccesAS2(evt:MouseEvent ) {
   // nom de la connexion définie dans AS2, nom de la fonction définie dans AS2
   emetteur.send("maConnexionAS2", "monActionProgrammeeAS2");
}
```

À l'intérieur de la fonction appelée par le gestionnaire d'événement en AS3, nous associons, à l'objet émetteur, la méthode send(). Cette méthode envoie une requête qui invoque le nom de la connexion (maConnexionAS2) et la fonction (monActionProgrammeeAS2), préalablement définies dans le fichier AS2.

En publiant le document, et en cliquant sur le bouton du document AS3, la liaison entre les deux fichiers est établie.



Il est possible d'appeler un document hébergé sur un autre serveur et d'un autre domaine que celui où figure le fichier appelant. Vous devez, pour le document appelé, ajouter l'instruction qui autorise le lecteur à communiquer avec un autre domaine. Pour cela, entre la définition de l'objet Local-Connexion et l'activation de la connexion, ajoutez une des deux instructions suivantes.

Si le fichier AS3 est sur un domaine différent mais identifié, inscrivez :

```
recepteur.allowDomain("http://www.nomdusite.com");
```

Ou bien, si le document AS3 est sur un domaine différent mais inconnu, inscrivez :

```
//recepteur.allowDomain("*");
```

À retenir

- Les documents AS1/AS2 peuvent être importés dans un document AS3 à l'aide d'un simple chargeur, mais aucune interaction n'est possible avec le document importé.
- Pour autoriser l'interaction du document AS1/AS2 chargé, nous devons établir une connexion à l'aide d'un objet LocalConnexion. Le document AS3 peut alors invoquer des fonctions développées dans le fichier AS1/AS2, d'une ancienne version de langage.

Synthèse

Dans ce chapitre, vous avez appris à redistribuer les contenus à l'écran, en fonction des dimensions d'une fenêtre, et à projeter ces contenus dans un affichage de type Plein écran, tout en permettant à l'utilisateur de restaurer l'affichage initial pour la réalisation d'un site full Flash. Vous avez également appris à gérer l'affichage de la typographie, à l'intégrer si nécessaire et à la partager, ainsi que des symboles, avec d'autres documents. Vous savez aussi comment importer physiquement un document PDF pour la publication et éviter de sortir d'un univers tout en Flash. Vous avez enfin appris à interagir depuis un document AS3, avec un contenu développé en ActionScript 1 ou 2. Vous êtes à présent en mesure de réaliser des systèmes d'affichage sophistiqués et optimisés, qui valorisent les contenus en Flash, au-delà de la surface d'affichage astreinte aux dimensions utiles d'une fenêtre de navigateur, toutes versions de Flash confondues.

Les solutions de référencement du Flash

Introduction

Flash a pendant longtemps été réputé pour offrir un contenu totalement opaque pour les moteurs de recherche. Mais, voici la chose révolue. Google et Adobe collaborent maintenant pour favoriser la reconnaissance des contenus édités au format Flash.

Depuis le printemps 2008, tout contenu Flash structuré selon les recommandations de Google est officiellement enregistré et présenté de la même manière que tout autre contenu HTML, aussi stricte en soit la mise en forme. Nous détaillons, dans ce chapitre, les recommandations émises par Google pour favoriser l'indexation d'un contenu réalisé en Flash.

Nous abordons aussi d'autres aspects comme : le *pageRanking* qui révèle la pertinence d'un site en fonction de sa popularité autant que par son contenu ; les métadonnées XMP ajoutée de manière intrinsèque aux contenus Flash ; l'intégration XHTML stricte plus favorable à la lisibilité d'un contenu Flash ; et l'affichage de contenu HTML alternatif pour les utilisateurs qui ne disposent pas du lecteur Flash.

À l'issue de cette annexe, vous saurez rendre vos réalisations parfaitement visibles pour les moteurs de recherche. Vous saurez même comment vous positionner mieux que des contenus réalisés en HTML simple.

Intégration XHTML stricte

Les annuaires sont sensibles à l'accessibilité des contenus. La manière d'intégrer les fichiers Flash favorise donc le positionnement du site.

Lors de l'intégration d'un fichier SWF dans une page HTML, Flash et Dreamweaver utilisent les balises <embed> et <object>. La balise <embed> ne fait pas partie des spécifications des normes strictes d'intégration XHTML. Pour intégrer le Flash proprement, il est préférable d'employer uniquement la balise <object>.

Le code de base employé pour une intégration stricte du Flash est :

Il est à préciser que la balise <object>, employée seule, active aussi un blocage dans Internet Explorer qui demande à l'utilisateur de confirmer l'activation du contenu en Flash avant de poursuivre sa lecture (comme contrôle Active-X). D'autres solutions d'intégration plus

souples sont disponibles pour Flash. Nous les abordons en fin d'annexe, avec l'utilisation du kit *SWFObject*2.

À retenir

- L'intégration stricte de Flash avec la balise <object> favorise indirectement son indexation auprès des moteurs de recherche.
- L'utilisation de la balise <object> implique un blocage dans Internet Explorer.
- D'autres solutions d'intégration sont disponibles avec SWFObject2.

Structurer un document pour l'API de Google

Depuis juin 2008, Google référence officiellement le Flash (lire http://googlewebmaster-central.blogspot.com/2008/06/improved-flash-indexing.html). Les résultats des pages de recherche démontrent que le contenu véhiculé dans un document Flash est désormais bien pris en compte (voir Figure 16.1).

Figure 16.1 Indexation d'une animation SWF avant la nouvelle API de Google Comparaison des résultats de recherche Google, avant et après 2008. Deep Impact www.jpl.nasa.gow/multimedia/deep-impact/index-flash.html - 3k - Cached - Similar pages Indexation d'une animation SWF depuis la nouvelle API de Google après 2008. Deep Impact NASA's Hubble, Spitzer and Chandra Space Telescopes will be recording these ... ORBIT PATHS This animation shows the trajectory of Deep Impact and the orbit ... www.jpl.nasa.gow/multimedia/deep-impact/index-flash.html - 3k - Cached - Similar pages

Pour que les documents Flash puissent être correctement indexés, ils doivent obéir aux spécifications suivantes :

- Tous les textes contenus dans un document SWF sont enregistrés, dès lors qu'ils sont visibles à l'écran et édités dans un champ de texte statique, dynamique ou de saisie. Les textes placés dans des symboles, dans des boutons sont également pris en compte. Seuls les textes images ou vectorisés ne sont pas identifiés.
- Les URL affichées dans le texte, si elles sont cliquables, sont également enregistrées.
 Elles peuvent être actives à partir des fonctionnalités de lien définies depuis l'Inspecteur de propriétés, ou en ActionScript, toutes versions du langage confondues.
- Les fichiers SWF, XML, HTML et textes externes, liés à un contenu Flash identifiable (un texte) sont également enregistrés. Important, Google fait même ressortir la page HTML qui contient le Flash. Donc, non seulement Flash n'est pas opaque pour les moteurs, mais l'utilisation de la technologie Flash favorise l'indexation d'une page HTML.
- Les images, les vidéos et les graphismes vectoriels ne sont pas, en revanche, enregistrés.

- Si les contenus image et formes graphiques ne sont pas identifiables par Google, utilisez ces formats pour véhiculer des informations au demeurant inutiles, comme le texte "chargement en cours", par exemple.
- Les documents Flash intégrés *via* une instruction JavaScript ne sont pas référencés. JavaScript apparaît comme un mur anti-référencement. Préférez une intégration stricte avec la balise <object> (voir section précédente). Mais cela provoque un blocage d'Internet Explorer. Optez plutôt pour une indexation qui se base sur un contenu alternatif (voir fin d'annexe).
- Flash distingue les pages HTML, les textes TXT et les fichiers XML appelés par le SWF du fichier SWF lui-même. Ces contenus risquent donc d'être présentés en accès parallèle aux animations Flash. Veillez à désigner clairement leur libellé pour éviter les confusions et orienter autant que possible les utilisateurs.
- Les textes bidirectionnels ne sont pas encore indéxés, mais Google y travaille (Hébreu, Arabe).
- Ne cachez pas d'informations sans rapport avec les noms et les descriptions affectés aux objets (technique du spamdexing ou spam). Le fonctionnement de Google favorise toujours l'indexation des sites qui servent l'utilisateur. Tout comportement mal veillant nuisible pour l'utilisateur est donc également nuisible pour votre positionnement.
- Une organisation sémantique de la structure du site (noms des fichiers, noms des dossiers) est primordiale. Aussi, attribuez des noms de fichiers et de dossiers en rapport avec le thème abordé, naturellement, sans caractères spéciaux, ni espaces. Les tirets sont considérés par Google comme des séparateurs sémantiques, mais ils ne sont pas toujours bien supportés par l'API de Flash dans le cas de documents SWF imbriqués.
- Vous pouvez également ajouter des métadonnées aux fichiers SWF afin d'orienter le robot. Pour en savoir plus sur cette technique, reportez-vous à la section suivante.
- Utilisez une mise en page XHTML claire pour le robot, qui lui permettra de distinguer sémantiquement le positionnement du contenu en Flash des autres contenus. Pour cela, utilisez des balises div dont les identifiants se rapportent aux données véhiculées (Par exemple, utilisez un conteneur HTML <div id="flash">...</div> pour contenir le Flash et <embed id="animation" ... /> pour afficher l'animation Flash).
- Pensez à toujours associer un contenu alternatif complémentaire au format HTML. Cela peut être une note d'intention du site, un résumé, une version mobile, un document PDF. Tout texte qui accompagne le Flash aide à positionner le ou les fichiers présentés.
- Vous pouvez également élaguer les éléments inutiles qui ne doivent pas être pris en compte lors de l'indexation. Filtrez pour cela les contenus utiles en utilisant un fichier "robot.txt" (consultez l'adresse http://www.google.com/support/webmasters/bin/answer.py?hl=fr&answer=156449 pour en savoir plus sur l'utilisation du fichier robot.txt).
- N'hésitez pas à consulter: http://www.google.com/support/webmasters/bin/answer.py?hl=fr&answer=72746 pour en savoir plus.

Pour évaluer au mieux la manière dont votre projet peut être interprété par Google, pensez que Google favorise toujours les contenus au service de l'utilisateur. Si vous apportez des contenus utiles, bien organisés, faciles d'accès, ils seront mis en avant, même en Flash. Si vous tentez de manipuler l'utilisateur, d'agir à son insu, de forcer certains contenus, de cacher des textes en les mettant blanc sur fond blanc, cela pénalisera votre positionnement. Google se met toujours à la place de l'utilisateur pour vérifier un contenu en agissant dans l'intérêt de l'utilisateur. En respectant cette logique, vous améliorez sensiblement le positionnement de vos réalisations.



Le pageRanking. Le pageRanking désigne la popularité d'un site et principalement, la qualité des liens extérieurs qui y font référence. Cette donnée représente 50% des critères qui aident Google à déterminer le positionnement d'un site dans ses pages de résultat. Il en résulte que, même un site entièrement opaque et pour lequel aucun effort d'optimisation n'est déployé, parvient très bien à apparaître en tête du placement pour une requête ciblée. Un site 100 % HTML strict ne garantie donc pas, à lui seul, un positionnement meilleur qu'un site tout en Flash, même opaque. Néanmoins, nous ne saurons que vous conseiller de favoriser l'accessibilité de votre projet et son optimisation, afin de répondre au mieux aux différents besoins des utilisateurs.

Une excellente illustration de l'effet du pageRanking est cette anecdote apparue pendant les élections présidentielles de 2007, en France. Une équipe de webmasters avait placé au sein de nombreux blogs et forums de discussion, des hyperliens qui pointaient vers une biographie du personnage Iznogoud, personnage, ô combien réputé pour son goût prononcé du pouvoir. Ironie de l'histoire, en associant au libellé de ce lien, l'expression "Nicolas Sarkozy Président", ces webmasters ont prouvé la toute puissance de Google. Quelques semaines plus tard, il suffisait en effet de saisir "Nicolas Sarkozy Président" dans le moteur de Google, pour que la première page proposée soit la fiche biographique du personnage Iznogoud. Le mécanisme ayant opéré, ils n'avaient pas tardé à proposer l'inverse : un lien de libellé Iznogoud qui pointait vers la page de candidature officielle du candidat Nicolas Sarkozy. Sans évidemment porter de jugement sur le fond de cette forme de blague contemporaine, le résultat obtenu a prouvé combien le mécanisme de Google prenait davantage en considération, le libellé et l'emplacement des liens sur l'Internet, que les contenus qui y sont associés, eux-mêmes. C'est la raison pour laquelle nous pouvons affirmer aujourd'hui qu'un site réalisé intégralement en Flash, pour lequel une campagne virale bien étudiée a été déployée, peut obtenir un positionnement bien supérieur au référencement naturel obtenu par un site XHTML strict (voir Fiqure 16.1).



Le débat Flash vs. XHTML. Flash ne se positionne pas comme une technologie concurrente à la création de sites XHTML stricts. Ces techniques sont parfaitement complémentaires, au contraire. L'une apporte l'accessibilité (HTML), l'autre la richesse et les fonctionnalités (Flash). À titre d'exemple, une entreprise pourra réaliser un site 100 % XHTML pour offrir un service générique et institutionnel pérenne qui véhiculera son image "corporate", une forme de "service public" en somme. Elle peut l'accompagner, plus ponctuellement et régulièrement, sur des périodes plus courtes, avec des mini sites événementiels tout en Flash. Cela lui permet, tout en apportant un service de base, élémentaire, de valoriser et mieux communiquer aussi sur ses produits en proposant des fonctionnalités richmédias irréalisables en XHTML. Les deux conceptions se complètent donc parfaitement. Il n'y a pas de camp tout désigné qui puisse se venter d'offrir une solution meilleure qu'une autre. Chaque technologie apporte une valeur ajoutée utile et complémentaire l'une de l'autre.

À retenir

- Google sait enregistrer les textes contenus dans un document SWF.
- Les fichiers qui accompagnent un document Flash, importés dans le SWF ou environnants (page HTML qui le contient) sont valorisés.
- L'intégration du Flash avec JavaScript favorise le blocage avec Internet Explorer. Cette solution permet, cela dit, de positionner le Flash à l'aide d'un contenu alternatif.

Les métadonnées des fichiers SWF

Les métadonnées des pages HTML sont des facteurs importants pour le référencement, même si ces derniers sont de moins en moins pris en considération, à la faveur des contenus eux-mêmes. Lorsque les métadonnées accompagnent les contenus, en intégrant des textes et des mots qui soulignent le sens réel des informations affichées à l'écran, elles valorisent le positionnement des pages.

L'ajout de métadonnées est désormais possible pour les documents SWF, comme nous le faisons pour les documents HTML. Depuis Flash CS4, il suffit de renseigner un tableau de descriptions codé au format XMP, puis, de publier le document, pour que les informations soient automatiquement intégrées à l'animation. Dès que le robot identifie le fichier SWF, il analyse ces métadonnées et affine le positionnement du document.

- Pour ajouter des métadonnées à un document Flash, dans Flash, faites Fichier > Informations.
- 2. La table des métadonnées s'affiche (voir Figure 16.2).

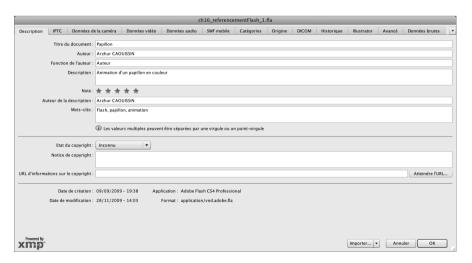


Figure 16.2Aperçu de la fenêtre Informations.

- 3. Dans cette fenêtre, différents onglets sont disponibles :
 - L'onglet Description permet de renseigner le profil général de l'auteur et d'informer les robots sur les caractéristiques générales du document.
 - L'onglet IPTC organise plus en détail les informations relatives à la signature du document, selon la norme conventionnelle internationale employée dans le secteur de la presse.
 - Les autres onglets permettent d'ajouter des informations en rapport avec d'autres systèmes d'indexation et d'autres formats de fichiers.
 - À l'exception de l'onglet SWF mobile, réservé pour la publication des documents Flash à destination des appareils mobiles, nous n'en aurons pas usage pour la création de sites web.
- 4. Renseignez les champs requis dans les différents onglets. Puis validez.
- 5. Publier le SWF en faisant Ctrl+Entrée (Windows) ou Cmd+Entrée (Mac).

Le fichier SWF contient ses propres métadonnées. Si vous disposez de la Creative Suite Adobe, vous pouvez lire ces données depuis Bridge. Naviguez alors jusqu'au fichier SWF pour lire les métadonnées (voir Figure 16.3).

Figure 16.3Affichage des métadonnées dans Bridge.





Il est possible de générer des métadonnées via ActionScript à l'aide de la méthode setMetadata(). Une documentation en ligne présente les commandes disponibles à l'adresse suivante : http://help.adobe.com/fr_FR/Flash/10.0_ExtendingFlash/
WS5b3ccc516d4fbf351e63e3d118a9024f3f-7dcf.html

À retenir

- Pour ajouter des métadonnées dans un fichier SWF, nous utilisons la fenêtre d'information du document Flash.
- Ces métadonnées sont lisibles par Google, mais aussi et entre autres depuis Bridge.

Gérer le contenu alternatif pour les robots et les appareils nomades

Les éléments XHTML (<div></div> ou <embed />) ne sont pas pris en compte par les moteurs lorsqu'ils sont affichés à partir d'une commande JavaScript. Pour rendre un contenu visible, il est donc préférable d'organiser son affichage directement à partir de ces balises XHTML sans les envelopper de comportements JavaScript (voir aussi http://fr.wikipedia.org/wiki/Web_profond). Mais, bien que le contenu affiché par une instruction JavaScript neutralise l'accès au document Flash, il offre un certain avantage.

À défaut de vouloir rendre le document Flash indexable pour l'API de Google, vous pouvez aussi afficher un contenu texte XHTML, alternatif au Flash. Dans ce contexte, c'est justement le contenu HTML alternatif qui va aiguiller le robot. Cette technique offre aussi un deuxième avantage. Nous pouvons afficher un contenu alternatif au Flash à destination des appareils nomades non équipés du lecteur. Cela permet, par exemple, de proposer un service différencié et davantage ciblé pour les utilisateurs d'appareils nomades. Certains appareils continuent, en effet, de ne pas pouvoir lire le Flash. Cette section apporte les solutions alternatives pour ces utilisateurs.



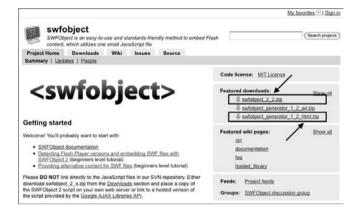
Les contenus alternatifs intelligents. Par le passé, nous développions un clone du site Flash, en guise de contenu alternatif. Une aberration qui a été repensée avec l'arrivée des appareils mobiles et la vague du Web 2.0. Lorsque vous utilisez un support différent d'un ordinateur classique (téléphone mobile), vous attendez un contenu approprié et qui réponde au mieux à votre besoin au moment où vous consultez. Il paraît donc plus cohérent de cibler effectivement l'affichage des contenus en fonction de la plateforme qui l'exécute et non plus de le doubler de manière parfaitement stérile. Si le site doit être consulté sur un ordinateur, il peut offrir une navigation riche, de nombreuses fonctionnalités parfois gourmandes en ressources, sans trop de difficulté. Mais si le site doit pouvoir être visité rapidement, entre deux rendez-vous, avec une connexion payante à la minute, sur un écran de taille réduite et sur un appareil aux capacités limitées, en plein soleil, vous ne pouvez évidemment pas imposer le même service que sur le site de base. Pensez à utiliser la technique JavaScript qui distingue le contenu Flash d'un contenu HTML pour permettre également d'offrir un contenu mobile approprié, différent du site référent développé en Flash.

Installer le kit SWFObject2

Pour réaliser une page HTML conforme, qui ne provoque pas le blocage d'Internet Explorer, et qui permette de spécifier à partir de quelle version du lecteur Flash, nous préférons afficher un contenu alternatif, tout en offrant un contenu identifiable pour Google, nous utilisons le kit de développement *SWFObject2*.

Pour utiliser ce kit facilement, nous téléchargeons le script SWFObject et son installeur. Le script est à placer dans le répertoire du site. L'installeur est une page HTML/JavaScript qui génère, à la volée, le code HTML requis pour une intégration conforme de l'animation Flash. Ce kit est disponible gratuitement à l'adresse suivante : http://code.google.com/p/swfobject/. Téléchargez les fichiers "SWFObject2_2.zip" (le script) et le fichier "SWFGenerator_1_2_html.zip" (le générateur) (voir Figure 16.4).

Figure 16.4Téléchargement du kit
SWFObject2.



Dans le répertoire de votre site (en l'occurrence, dans notre dossier Exemples), décompressez directement les deux fichiers. Le dossier "SWFObject.js" peut être placé à la racine du projet. La page HTML, index.html, une fois utilisée, peut être supprimée du projet. Dans les fichiers d'exemple de cet ouvrage, nous avons renommé cette page "SWFObject-Index.html", pour éviter toute confusion avec d'autres fichiers.

En parcourant le dossier "SWFObject/" apparu à la décompression, nous distinguons, dans ce répertoire, le fichier JavaScript "swfobject.js", ainsi qu'un installeur de mise à jour du lecteur Flash nommé "expressInstall.swf". Dans ce même dossier, deux pages HTML proposent déjà le code pour l'intégration d'un document Flash selon deux méthodes différentes. La page "index.html" affiche une intégration stricte et conforme avec des balises HTML. La page "index_dynamic.html" intègre le Flash à partir du JavaScript. Cette seconde proposition adopte un codage tout à fait conforme, mais elle offre l'avantage de neutraliser le blocage d'Internet Explorer sur les applications multimédias (avec le message "cliquez pour activer ce contrôle"). Vous pouvez donc utiliser l'un ou l'autre de ces deux fichiers pour mettre en forme votre page HTML, selon la technique de votre choix.

Vous pouvez aussi lancer dans le navigateur la page que nous avons renommée "SWFObjectIndex.html", pour obtenir un code personnalisé. Pour utiliser cet installeur, procédez comme suit :

- 1. Lancez la page SWFObjectIndex.html dans un navigateur.
- 2. Cette page affiche un formulaire dans lequel nous avons simplement à désigner le nom et l'emplacement relatif du document Flash (à partir de la page HTML qui doit contenir le Flash), ainsi que ses dimensions en pourcentage ou en pixels (voir Figure 16.5).

Figure 16.5

Aperçu du générateur dans le navigateur.

rena (*) induses required	Feld
SWFObject configuration	= (=)
SWFORGER CJUL 1	swfobject.js
Publishing method: 1	Name publishing (B) what is this?
Detect Flash version: 1	9 0 0
Adobe Express Install:	expressinstall.swf
-SWF definition [;]	
Flash Cswfi: "	untitled self
Dimensions: 1	600 x 600 greats 128
Flash content id "	myPlashContent
mera.	
HTML defension [;]-	100 march 200 ma
HTM, Template	XMTML 1.0 SHAD (B)
Character encoding:	10-8819-1 (3)
Alternative content	what is this?
clas size, to	.abdok.org/pd/pdf13abgkd/ppdf'> T132//nowr.bDobe.com/langes/phared/downloam_buttoms/get_fishh_player.gif' 41t*'bet.Abdo
Canassian (Class)	
Generated output [;]-	
Generated output [;]	
-Generated output [_] -	

- 3. Le générateur affiche les noms des fichiers scripts (swfobject.js) et de l'installeur express (expressInstal.swf), comme figurant à la racine du projet.
- 4. Pensez d'abord à ajouter le nom du répertoire swfobject/ qui organise ces éléments, afin que le navigateur qui exécutera la page puisse atteindre ces fichiers.
- 5. Choisissez un type d'intégration stricte ou dynamique.
- 6. Cliquez ensuite sur Generate afin d'obtenir le code HTML de base de votre document.
- 7. Puis, sélectionnez ce code et copiez-le dans une nouvelle page HTML vierge que vous nommerez index.html, si vous en faites votre page d'accueil (voir Figure 16.6).

Figure 16.6
Paramétrage du document HTML.

noni r'i return muirel SWFObject configuratio	
SWFObject configuratio	TWO II
	m(±)
SWFObject (jsl: *	xwfotbject/xwfotbject.liv
Publishing method: 1	Oynamic publishing (B) what is this?
Detect Flash version: "	10 0 0
Adobe Express Install:	M perfolipect/expressinatall-4
HTML container id. "	accesthobile, what is this?
SWF defection (_)	
Flash Coeff: 1	(b.15. referencement/lash
Omensions: 7	100 x 100 [percentage (B)
Flash content id	contenu
mere	
HTML definition [_]	
HTML Templane:	SHTML LO MYKE (E)
Character encoding:	no-8819-1 (\$)
Alternative content:	what is this?
	10 "-//WW//VFD MRTHL 1.0 DEFET//BB" "HELDI//WWW.WS.SEG/TR/HESELL/UTS/HESELL-
Constitute output 2 1 1 1 1 1 1 1 1 1	Li "(real/c/mb arms. Li a micin/mb "magn/convolveng/mb/mami/mc/magn/mb/mami/mc/mb/mb/mb/mb/mb/mb/mb/mb/mb/mb/mb/mb/mb/
Cenerated output 2 1 1 1 1 1 1 1 1 1	LR(rest/c/mb arms. La missi/c/mb -/mag/c/mc/siss/mb/masi/res/masi/mb/masi/res/masi/mb/masi/res/masi/mb/masi/res/masi/mb/masi/res/masi/mb/masi/res/masi/mb/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/masi/res/masi/r

Dans le dossier des exemples du livre, vous trouverez la page enregistrée, avec l'option d'intégration dynamique, sous le nom de "ch16 referncementFlash 2.html".

En l'ouvrant dans le navigateur, si le lecteur Flash 10 ou supérieur est intégré, la page apparaît normalement. À défaut, la version alternative est affichée. Cette version alternative est basique. Le contenu Flash s'affiche dans un coin de la page. Pour améliorer l'intégration des contenus, nous pouvons personnaliser le document.



Tester avec plusieurs versions du lecteur Flash avec Flash Switcher. Vous pouvez tester le comportement de la page à partir de différentes versions de lecteur Flash, depuis la même fenêtre de navigateur, sans nécessairement disposer de plusieurs configurations pour ce faire. L'extension de navigateur *Flash Switcher* offre cette possibilité. Pour en savoir plus l'assistant Flash Switcher, consultez le lien suivant : http://www.sephiroth.it/firefox/flash switcher/index.php.

Personnaliser le document

Dans le mécanisme du JavaScript SWFObject, le contenu alternatif est toujours affiché par défaut dans la fenêtre du navigateur. C'est uniquement lorsque JavaScript détecte le lecteur Flash qu'il affiche l'animation SWF (voir Figure 16.7). Le cas échéant, c'est la version HTML qui est affichée. Mais nous devons bien comprendre que ce HTML est déjà en place, il n'est simplement pas visible si le lecteur Flash est actif (voir Figure 16.8). Pour cette raison, ce mécanisme est favorable à l'indexation du contenu et peut être interprété par tous les navigateurs.

Figure 16.7 Aperçu de l'animation Flash.



Figure 16.8

Aperçu de la page HTML alternative.





Exemples > ch16_referencementFlash_2b.html Exemples > ch16_referencementFlash_2.fla

Dans le document "ch16_referencementFlash_2b.html", nous avons substitué le contenu alternatif affiché par défaut par le générateur SWFObject (voir section précédente), par un texte formaté avec des styles CSS, une image, un hyperlien et des titres.

Dans le corps de la page, en lieu et place du contenu de la balise <div id="content"></div>, nous avons introduit le texte alternatif. Les formatages CSS, eux, sont ajoutés dans l'en-tête du document :

```
<style type="text/css">
<!--
html {
  height: 100%;
body {
 padding:0px;
 margin:0px;
  font-size: 16px;
  font-family: Arial, Helvetica;
  background-color: #071B22;
  color: #FFF;
  background-image: url();
  background-repeat: repeat-x;
  height: 100%;
  overflow: hidden;
#donnees{
  height: 90%;
  width:550px;
  margin: 20px;
```

```
position: relative;
  text-align: justify;
h1{
  color: #FFF;
  font-size:48px;
h2{
  font-size:36px;
  border-bottom-width: 1px;
  border-bottom-style: solid;
  border-bottom-color: #FFF;
h3{
  display:block;
  font-size: 12px;
  background-color: #333;
  padding: 3px;
  text-transform: uppercase;
}
a:link {
  color: #CCC;
#design a:link {
  color: #666;
a:visited {
  color: #CCC;
a:hover {
color: #FFF;
#design a:visited {
  color: #666;
#design a:hover {
  color: #000;
ul, li{
  margin:0;
  padding:0;
li{
  display:inline;
#design {
  background-color: #FFF;
  color: #000;
  padding-top: 5px;
  padding-right: 10px;
  padding-bottom: 5px;
  padding-left: 10px;
  font-size: 14px;
#design h2 {
  font-size:18px;
  border-bottom-width: 1px;
  border-bottom-style: solid;
  border-bottom-color: #000;
}
```

Afin que le contenu Flash puisse apparaître en occupant toute la surface de la fenêtre du navigateur, nous lui avons attribué des dimensions de 100 % en largeur et en hauteur. Mais, comme nous employons un codage XHTML strict, nous devons aussi rappeler au navigateur que le corps du document et la structure HTML générale sont étendus, eux aussi, sur toute la surface de la fenêtre de navigateur. À défaut, le contenu Flash n'apparaîtra pas dans Firefox. C'est la raison pour laquelle, dans la feuille de style, nous ajoutons les attributs CSS height:100% pour chacun des éléments qui contiennent le Flash.

De même, afin de supprimer la barre d'ascenseur verticale qui résulte de l'intégration du Flash, nous la masquons en ajoutant, dans le style pour le corps de la page (body), l'attribut overFlow, passé sur la valeur "masqué" (hidden). Nous pourrions aussi afficher un Flash à seulement 99 % de la hauteur, au risque toutefois d'ajouter une petite marge en bas de la fenêtre de navigateur. Vous appliquerez l'option à votre convenance. Notez cependant que l'option overFlow induit une surface limite pour afficher le contenu alternatif. Dans ce cas, si un contenu important doit apparaître, nous vous conseillons de créer d'abord un sommaire, puis de répartir le contenu sur différentes pages HTML successives qui elles, n'utilisent pas cet attribut.

Le contenu SWF peut désormais être affiché à 100 %. Mais pour cela, nous devons aussi ajouter des contrôles ActionScript, comme vu Chapitre 15, afin de redéfinir le positionnement et l'échelle des objets selon les dimensions de la fenêtre. Sans cela, le Flash resterait calé en haut et à gauche, sans être redimensionné. Seule la couleur d'arrière-plan du document SWF occuperait toute la surface de l'écran.

Le document Flash "ch16_referncement_2.fla", dont l'exportation au format SWF est intégréé dans notre page HTML, affiche le code suivant :

```
//---- initialisation
import gs.*;
import gs.easing.*;
import gs.events.*;
//---- actions
interface_mc.lancer_btn.addEventListener(MouseEvent.CLICK,volPapillon);
function volPapillon (evt:MouseEvent) {
 TweenMax.to(interface mc.Papillon mc, 3, {rotation:-360, delay:0.2,
 ease:Strong.easeInOut});
// mise à l'échelle pour remplissage à 100 % de la page HTML
stage.scaleMode = StageScaleMode.NO SCALE;
stage.align = StageAlign.TOP LEFT;
stage.addEventListener(Event.RESIZE, redimensionneSWF);
function redimensionneSWF(event:Event) {
 fond_mc.width=stage.stageWidth;
 fond mc.height=stage.stageHeight;
 interface mc.x=stage.stageWidth/2;
 interface mc.y=(stage.stageHeight/2)-20;
```

Ce code réorganise l'affichage et les dimensions des éléments dans la scène, selon la taille de la fenêtre du navigateur.

Pour en savoir plus sur la gestion élastique et flottante du contenu Flash, reportez-vous au Chapitre 15 qui traite de la création de sites Full Flash.



Les pages satellites. Les pages satellites sont des pages HTML qui reprennent des contenus textes édités initialement dans un site en Flash. Cette technique, employée jusqu'à ce que Google sache interpréter le Flash (jusqu'en 2008 donc), aidait l'indexation des sites tout en Flash. Elle n'est plus recommandée aujourd'hui, parce que Google sait maintenant lire le contenu en Flash, mais aussi parce que, comme toute technique de diversion, elle finit malheureusement par être trop souvent employée contre l'utilisateur. Ces pages ne sont donc plus vraiment prises en compte sauf lorsqu'elles servent réellement le contenu et se présentent comme des pages d'accessibilité alternatives, justement. C'est le cas pour des textes destinés aux appareils mobiles ou qui apportent des informations utiles accessibles rapidement au format texte, par exemple (adresses, coordonnées, contenu éditorial).



Pour en savoir plus sur SWFObject2. Une page française décrit admirablement les options disponibles pour la gestion d'un contenu Flash avec SWFObject2. Consultez ce lien clair et instructif pour en savoir plus sur ce sujet : http://egypte.olympe-network.com/swfobject-francais.html.

À retenir

- Il est possible d'afficher un contenu alternatif à Flash pour les utilisateurs qui ne disposent pas du lecteur. Pour cela, nous utilisons le script SWFObject2.
- Il existe deux méthodes conformes pour l'intégration du Flash : une méthode HTML stricte et une méthode en JavaScript pour contourner le blocage d'Internet Explorer.
- La gestion de l'affichage du Flash à 100 % de la surface de la fenêtre de navigateur se détermine en attribuant des hauteurs de 100 % aux conteneurs HTML, et en définissant des comportements d'échelle et de positionnement, en ActionScript, dans le Flash.

Synthèse

Dans cette annexe, vous avez appris à rendre indexable un site entièrement réalisé en Flash, pour favoriser son positionnement dans les moteurs de recherche. Vous avez appréhendé le mécanisme de Google dans sa recherche de contenus. Vous avez appris à gérer l'affichage d'un site pour les systèmes qui ne disposent pas du lecteur Flash et notamment certains appareils mobiles. Vous êtes en mesure de réaliser et concevoir des interfaces Flash entièrement indexables par les moteurs de recherche.

17

L'accessibilité dans Flash

Introduction

Dans cette annexe, nous présentons les outils d'aide à la conception de sites accessibles, dans Flash, pour les personnes à "mobilité réduite sur le Web", qui utilisent des logiciels d'accompagnement pour lire les contenus (lecteurs d'écran).

Si Flash est naturellement permissif aux lecteurs d'écran, à condition toutefois qu'il soit intégré de manière stricte, des options de formatage et des méthodes de conception permettent d'organiser le contenu de manière encore plus lisible.

L'organisation d'un site accessible apparaît bien souvent comme une contrainte pesante en production, mais elle permet souvent de résoudre aussi des problèmes évidents d'ergonomie. Si un site est accessible pour un lecteur d'écran, il le sera pour tout utilisateur. De plus, si un site est accessible et pertinent pour tout utilisateur, il sera aussi valorisé par les moteurs de recherche. Réaliser un site accessible offre donc de nombreux avantages qui apportent à tous les utilisateurs. Rien n'empêche cela dit, ponctuellement, de réaliser des contenus qui sortent de ce cadre drastique, si la base, elle, demeure au moins suffisamment ouverte à tous les utilisateurs.

Flash propose, dans son interface, différents outils d'aide à l'accessibilité. Les métadonnées, que nous avons développées précédemment, y concourent. Mais aussi, les titres, les descriptions, l'ordre des tabulations et des champs, ainsi que quelques classes ActionsScript dont nous vous présentons ici les ressources pour pouvoir les utiliser et étendre plus spécifiquement vos développement vers ce type de configuration.

Les options d'accessibilité sont utiles pour toutes les personnes "à mobilité réduite sur le Web" au sens large, qu'il s'agisse de malentendants, non-voyants, paraplégiques, mais aussi les "accidentés ponctuels de la vie domestique" voire, les utilisateurs sans handicap mais dont l'équipement informatique ne fonctionne tout simplement pas correctement. Un étranger sur un site exclusivement en français est en situation de handicap. Une personne sans plugin adéquat pour visualiser un site Flash est en situation de handicap (voir Annexe précédente). Un texte écrit trop petit face à une personne malvoyante (et qui n'use pas de lecteur d'écran) et dont il n'est pas possible d'augmenter la taille du texte... est en situation de handicap. Une personne habituée à naviguer en usant que du clavier face à une interface utilisateur exclusivement basée sur l'usage de la souris est en situation d'handicap. Un site faisant appel à une vidéo sans sous-titre, pour un utilisateur en entreprise qui n'a pas d'haut parleur, le place en situation de handicap, Un site qui prévoit le chargement d'un PDF pour palier à un soucis d'accessibilité et qui fournit un PDF non accessible place l'utilisateur en situation d'handicap. Et autant d'autres situations cocasses qui nous mettent tous en situation de handicap. Nous sommes donc *a priori* tous concernés.

La fenêtre Accessibilité

Vous pouvez ajouter des titres, des descriptions et associer des contrôles de navigation, à tout symbole distribué dans Flash. Il suffit de renseigner les champs mis à disposition par la fenêtre d'accessibilité, pour ce faire.



Exemples > ch16_accessibiliteFlash_1.fla

Dans ce document, la scène affiche quelques symboles dont un champ de texte, un Movie-Clip et un bouton (voir Figure 17.1).

Figure 17.1Aperçu de la scène principale.



Vous pouvez ajouter des titres et des descriptions sur la scène et sur les objets eux-mêmes avec le menu Fenêtre > Accessibilité.

Une fenêtre apparaît et affiche différents champs de saisie. Cliquez en dehors de la scène de manière à ne rien sélectionner. La fenêtre d'accessibilité affiche alors les entrées pour la scène.

Renseignez les champs Nom et Description pour informer l'utilisateur sur le type de contenu mis en scène (voir Figure 17.2).

Vous pouvez également renseigner des informations, plus ponctuellement, pour chaque objet. Il suffit de le sélectionner et de renseigner les champs qui correspondent à votre sélection active.

Cliquez successivement sur les symboles interface_mc, papillon_mc, sur le champ de texte dynamique et sur le bouton, puis, inscrivez les textes qui décrivent succinctement ce qu'ils représentent.

Figure 17.2

Aperçu des options pour la scène principale.



Figure 17.3

Informations d'accessibilité du symbole papillon mc.

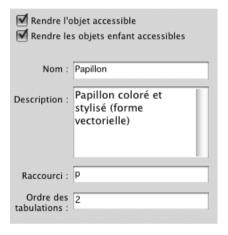


Figure 17.4

Informations d'accessibilité du champ de texte statique.



Figure 17.5
Informations
d'accessibilité du
symbole bouton.

Rendre l'o	bjet accessible
Nom :	bouton Lancer
Description :	Active une rotation à 360° sur le papillon
Raccourci :	I
Ordre des tabulations :	1

Lorsque des symboles enfants sont distribués à l'intérieur d'un objet, l'option Rendre les enfants accessibles est proposée. Par défaut, tous les enfants sont accessibles. Mais, dans certains cas, il peut être intéressant de neutraliser cette option afin d'éviter des bruits visuels inutiles et masquer des objets purement graphiques utilisés pour l'habillage. Cela contribue à simplifier l'ossature du document et facilite l'accès aux autres contenus.

Vous remarquez que, dans certains cas, les options Ordre des tabulations et raccourci sont également disponibles.

Les raccourcis permettent à l'utilisateur du lecteur d'écran d'accéder directement à l'objet, par simple activation d'une touche du clavier. Vous devez inscrire la lettre que vous souhaitez désigner comme touche à activer pour rendre cet objet accessible. Utilisez de préférence une lettre à consonance dure qui résume la tonalité du mot ou sa première lettre. Dans notre exemple, la lettre l est employée pour désigner le bouton Lancer.

Dans le champ Ordre des tabulations, spécifiez le nombre de fois pour lequel l'utilisateur doit appuyer sur la touche Tabulation, pour activer l'élément désigné. Dans notre exemple, le bouton Lancer est associé au chiffre 1. Il faudra donc appuyer une fois sur la touche Tabulation pour activer ce bouton, une fois le module Flash préalablement activé. Lorsque vous publiez la page HTML du document Flash dans le navigateur, pour activer l'animation avec les options d'accessibilité, procédez comme suit :

- 1. Activez d'abord l'objet Flash en cliquant dessus ou en appuyant sur la touche Tabulation une première fois.
- 2. Puis, une fois le Flash actif, appuyez de nouveau sur la touche Tabulation, autant de fois que la valeur désignée dans le champ Ordre des tabulations de la fenêtre d'accessibilité.
- 3. Puis, appuyez sur la touche Entrée pour confirmer l'activation des instructions associées à cet objet.

Afin de garantir une ergonomie la plus cohérente qui soit, utilisez un ordre des tabulations qui corresponde autant que possible à l'ordre logique et chronologique d'exécution des événements.



Accessibilité dans Flash. Pour en savoir plus sur l'accessibilité dans Flash, consultez aussi le lien suivant : http://help.adobe.com/fr_FR/Flash/10.0_UsingFlash/WSd60f23110762d6b883b18f10cb1-fe1af6-7c3ea.html#WSd60f23110762d6b883b18f10cb1fe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.html#WSd60f2310cffe1af6-7c3ea.ht

À retenir

- L'édition de textes de description pour l'accessibilité est disponible dans Flash. Pour cela, utilisez la fenêtre Accessibilité.
- Pour restreindre l'accès à des informations utiles uniquement, il est possible, à l'inverse, de neutraliser des objets. Pour cela, nous désactivons les options d'accessibilité des objectifs actifs, depuis la fenêtre d'accessibilité

Concevoir un document Flash pour les non-voyants et malentendants

Parmi les utilisateurs de lecteurs d'écran non-voyants ou malentendants, des recommandations spécifiques sont à suivre. Nous les détaillons dans cette section :

- Évitez de créer des boucles d'animation qui provoquent un bavardage encombrant. Lorsque la tête de lecture rejoue perpétuellement des images sur lesquels des textes sont affichés, ces informations peuvent être répétées indéfiniment et à voix haute, par les lecteurs d'écran. Cela perturbe la navigation. Pensez à appliquer un stop sur chaque image du scénario qui désigne une nouvelle rubrique. Privilégiez les conceptions de site dans le scénario, où chaque rubrique est jouée l'une à la suite de l'autre, et séparée par un stop. Les interfaces dynamiques ne sont pas encore totalement digestes par tous les lecteurs d'écran.
- Évitez d'imposer un son ou une ambiance musicale trop prégnante afin de permettre aux utilisateurs d'entendre leurs propres lecteurs d'écran lire les contenus.
- Ne vectorisez pas le texte, car les objets graphiques et les images ne peuvent pas être identifiés.
- Associez la plupart des actions à des commandes du clavier, en plus des commandes de souris. Cela permet aux personnes qui n'accèdent pas à la souris de compenser en utilisant le clavier.
- Le mode transparent et opaque, sans fenêtre, défini dans les paramètres HTML du document Flash, rendent le Flash complètement inaccessible. Évitez d'utiliser ces options.
- Pensez à associer, autant que possible, des textes à chaque objet interactif (boutons, liens, zones d'interaction) pour en faciliter la reconnaissance.



Aides au développement de contenus spécifiques pour les malentendants et les malvoyants. Des outils d'aide au développement sont disponibles à l'adresse suivante, pour Windows uniquement :

 $http://help.adobe.com/fr_FR/Flash/10.0_UsingFlash/WSd60f23110762d6b883b18f10cb1fe1af6-7c47a.html.\\$

À retenir

• Pour rendre un site accessible aux non-voyants et malentendants, il est recommandé d'organiser les contenus dans le scénario en privilégiant les textes.

Synthèse

Vous avez appris à optimiser un site pour l'accessibilité et rendre un contenu Flash presque universel. Vous êtes en mesure, à présent, de créer des sites pour le plus large public, tout en apportant une valeur graphique ajoutée et fonctionnelle, significative.

Ressources

Quelques ressources pédagogiques pour le développement en ActionScript 3 et les technologies associées à la création d'interfaces riches sont disponibles en ligne. Vous trouverez ci-après une liste non exhaustive des dernières publications qui complèteront idéalement cet ouvrage.

Livres

D'After Effects à Flash, de Flash à After Effects, Richard Harrington et Marcus Geduld. Éd. Pearson. DVD inclus. Couleur. 38 €.

Cet ouvrage vous accompagnera idéalement pour étendre vos connaissances sur la vidéo interactive avec des créations graphiques, réalisées dans After Effects.

L'art du bluff avec Flash CS4, Chris Georgennes. Éd. Pearson. DVD inclus. Couleur. 29 €.

Pour prendre en main les outils d'animation, de manière didactique, avec des notions pratiques d'animateurs professionnels.

Apprendre à programmer en ActionScript 3, Anne Tasso. Éd. Eyrolles. Noir et blanc. 29,90 €.

Ce livre aborde l'interfaçage dynamique sans aucun élément placé sur la scène. Il peut vous accompagner dans l'automatisation de certains processus de mise en forme, surtout pour des contenus externalisés.

Pratique d'ActionScript 3, Thibault Imbert. Éd. Pearson. Noir et blanc. 56 €.

La référence de la programmation en ActionScript 3. Ce livre contient les techniques les plus avancées en terme de développement, avec de nombreux conseils relevant de l'optimisation des ressources utilisateur. Vous y abordez aussi les entrailles du lecteur Flash pour vous familiariser encore plus avec la logique intrinsèque de Flash.

Tutoriels vidéos

www.alltutorial.com

Ce nouveau petit site pratique et prometteur propose des tutoriels vidéos entièrement gratuits et de qualité, en langue française, sur les logiciels graphiques.

Tv.adobe.com/fr

Ce site propose des tutoriels vidéos sur les produits Adobe, en français. D'autres tutoriels sont disponibles sur la version américaine, directement à l'adresse **tv.adobe.com**.

Sites web et forums

www.thefwa.com

Le site des références en Web design offre plusieurs centaines de liens vers les sites les plus élaborés graphiquement. Une référence incontournable.

flash.mediabox.fr

Le forum des développeurs en ActionScript. De nombreuses astuces et conseils sur vos développements.

blog.greensock.com/tweenmax

Les dernières mises à jour de la classe TweenMax, disponible directement sur ce site, et son forum, en anglais.

www.sephiroth.it

Un site communautaire sur le développement en général, dont ActionScript 3 en particulier.

sketchup.google.com/intl/fr

Le site de présentation du logiciel gratuit Google Sketchup, pour la modélisation 3D. Ce site propose aussi une bibliothèque d'objets 3D ainsi qu'une riche documentation, en français.

flash.mediabox.fr

Le forum des développeurs en ActionScript. De nombreuses astuces et conseils sur vos développements.

www.papervision3d.org

Le site de la communauté des développeurs de la classe PaperVision.

www.adobe.com/fr/devnet/actionscript

Le site communautaire des développeurs ActionScript, par Adobe. De nombreux conseils, méthodologies et solutions connexes y sont présentées.

www.yazo.net

Un site pour apprendre les bases du langage ActionScript 1, 2 et 3, avec les fichiers en libre téléchargement et la palette Yazo.

Dictionnaires

Aide contextuelle

Flash intègre une aide contextuelle pour vous aider à découvrir les définitions des différentes méthodes, propriétés, et autres éléments du langage ActionScript. Depuis la fenêtre d'Actions, faites clic-droit, puis activez l'option Aide.

www.adobe.com/support/documentation/fr/flash

Cette page recense les différentes aides disponibles pour l'apprentissage d'Action-Script, au format texte.

Index

3D 289 affichage, optimiser 268	Α	Adobe Premiere Pro, exporter en FLV 154
After Effects 147	44G 1 100	
améliorer l'affichage des	AAC, codec 183	ADSL 151, 164
images 250	Accélération 18, 53	Affichage
animation	matérielle 441	élastique 443
Caurina 250	Accessibilité 477, 481	flottant 443
Tween 250	Acrobat 450	plein écran 437, 440
TweenMax 250	ActionScript 134	AllowFullScreen 439
banque d'objets 3D libres 300	différence entre AS2 et AS3 9	quitter 443
champ de vision 284	importer AS1/AS2 dans AS3	Afficher
exporter 299	458	contenu alternatif à Flash 469
focale 284	philosophie du langage 9	plein écran 473
galerie vidéo 268	random 101	symbole 37
Google Sketchup exporter 300	addASCuePoint 226	Aide de Flash 111
livre interactif réaliste 250	addChild 275, 286	Aliasing 250, 268, 452
mouvements		Voir Lisser les images
d'objets 311	addpage() 354	align, propriétés 446
de caméra 304	Adobe After Effects 147	AllowFullScreen 440, 443
mur d'images 260	exporter	alpha 396
native 241	directement le projet vers	propriété 12
optimiser un site 3D 314	Flash 153	AMFPHP 135
PaperVision	FLV 149, 154	
aide en ligne 328	Adobe Media Encoder 177	Amortissement 18, 22, 53
guide de référence 328	audio, onglet 183	Anaglyphe Voir Relief
interactivité avec les	bandes noires 160	Angle 22
objets 3D 328	débit 183	Animation 7
lumière 328	exporter en FLV 155	3D (Voir 3D) 279
point de fuite 283, 284	Filtres (onglet) 162	boucle 23
propriété	formats pris en charge 156	détecter la fin 35
rotationX 241	fréquence 183	enchaîner 36, 47
rotationY 241	Multiplexeur 162	fluidité 28
rotationZ 241 x 241	niveau 182	forme 86
	profil 181	intervalles dans le temps 59
y 241 z 241, 279	recadrer 158	optimiser 14
rendu (publication) 311	réglages d'exportation 158	panoramique 7
verso des objets 3D 248	vidéo, onglet 179	plus nette 27
vidéo 174	Adobe Photoshop	rétablir les propriétés animée
Voir aussi Modéliser et	gestion des couches 365	après une interpolation 73
PaperVision	Voir aussi Photoshop	Tween et TweenMax 29

Animer un livre 3D 242	Bones 77, 79	chemin 103
API 331	Boucle 23, 59, 64	externe 103
Apple Compressor 147	accessibilité 481	erreurs 131
Apple Final Cut Pro, exporter	arrêter 64	Voir aussi Progression et
en Quick Time 154	for 258	Chargeur
Apple iMovie, exporter en DV	réaliser une texture raccord 24	Chargeur 235
ou MOV 155	Bouton	d'image 103
Apple Motion 141	bascule, interrupteur 442	Chronomètre 59
exporter en Quick Time 145	cibler son contenu 395	Ciblage
Apple Time Machine 279	conflit d'interactivité 400	contenu
Appliquer des propriétés de	états 389	imbriqués 232, 235, 237
MovieClip à un objet 64	interfaçable 395	inexistant 403
Armature 77	masquer la main au survol 414	symbole bouton 395
Array 384	neutraliser, désactiver 414	dynamique 259
Arrêter le scénario 58	bufferTime 203	étiquette dans Flash 423
	byteArray 451	objet dynamique 258
Arrondir un chiffre 231		Cinema 4D R11 299
AS3AnimationSystem 29		Cinématique inverse 77
Ascenseur 54	C	Classes 11
masquer 475	Cache vidéo 203	après la compilation 42
Assistant TweenMax 42		AS3AnimationSystem 29
Astérisque (*) 33	cacheAsBitmap 335	Caurina 29
Atteindre une image du scénario	Cadence 10	centraliser 42
59	animations Flash 165	compiler 102
Audio 161, 167, 206	vidéo 138, 143, 165 modifier 165	contextMenu 417
AAC 183		Filters 57, 65
Débit 183	Calque, nommer 8	ik 77
Fréquence 183	Camera3D 311	getArmatureAt() 82 IKJoint 77
volume 205	CaptionButton 215	registerElements() 83
	Caractères	importer 11, 32, 41, 51, 290
В	manquants 435	installer 428
В	spéciaux 10	localiser sur le système 32
Barre	Carrousel 251	mécanisme 29
d'ascenseur 54	Carte interactive 38	Timer 57, 59, 60
	catch 133, 134	transitionManager 57, 59, 63
de chargement 106	Caurina 29, 250	Twease 29
BevelFilter 65, 70	CBR, VBR Voir Vidéo	Tween 29, 34
Bézier 42, 44	Centre des symboles 24, 247	TweenEvent 35
Bibliothèque	Chaîne de caractères,	TweenMax 29, 38
exporter pour ActionScript 383	conversion 121	classPath 42
partagée 454	Champ de texte dynamique 111,	Clavier
Bitmap 331, 385	431	accessibilité 481
BitmapData 334, 359, 385	Chapitrage 207	commandes inactives en mode
Blender 299	Chargement 102	plein écran 441
BlurFilter 65, 68, 340	contenu	interactivité 315

navigation 320	nuancier, créer 357	E
tableau des valeurs des touches	prélever 360	
323 Clin Vain Maria Clin	puits, créer 357	easing 32
Clip Voir MovieClip	CoverFlow 251	Échantillonner la vidéo pour
Codage différentiel 180	Crénelage 250, 268, 452	Flash 153
Codec vidéo 137	Voir Lisser les images	Écouteur 103
colorMatrix 343	CSS, dans Flash 430	arrêter 55
ColorMatrixFilter 345	CUE_POINT 226	placer 63
ColorTransform 360	CuePoints 211, 217	Effets 26, 57
Communication Flash/HTML	currentFrame 376	amortissement 22
417	currentTarget 129, 131, 392	fenêtre 149
Compilation 29, 33	Curseur de défilement 54	ralenti 21
Composant		reflet 254
FLVPlayBack 169	D	relief 377
paramétrer 174 ScrollBar 431	В	spéciaux 141, 147, 153
Compter	Débit 183	Élasticité Voir Affichage flottant
nombre d'images dans le	Décalage 53	Embed 463
scénario 245	Décélération 18	Enchaîner
nombre d'objets dans un	Défilant 48	des actions 266, 412
symbole 258	Déformation 13	les animations 35
1.016		
concat 346	Dégrouper 451	ENTER FRAME 52, 207, 228
concat 346 Concaténation 346	Dégrouper 451 Déplacer un symbole	ENTER_FRAME 52, 207, 228 Enveloppe vidéo 173
	Dégrouper 451 Déplacer un symbole (startDrag) 406	Enveloppe vidéo 173
Concaténation 346	Déplacer un symbole	Enveloppe vidéo 173 Ergonomie 48
Concaténation 346 condenseWhite 434	Déplacer un symbole (startDrag) 406	Enveloppe vidéo 173 Ergonomie 48 Erreur
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393	Déplacer un symbole (startDrag) 406 Détecter	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303 Convertir un objet 64	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437 displayState 442	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303 Convertir un objet 64 en MovieClip 64	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437 displayState 442 Document, organiser 8	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426 Event 20 Event.ENTER_FRAME 10, 14 Exécution, mode pour les
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303 Convertir un objet 64 en MovieClip 64 copyChannel 385	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437 displayState 442 Document, organiser 8 Données, gérer avec XML 121	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426 Event 20 Event.ENTER_FRAME 10, 14
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 contentLoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303 Convertir un objet 64 en MovieClip 64 copyChannel 385 Couche, extraire 385	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437 displayState 442 Document, organiser 8 Données, gérer avec XML 121 draw 385	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426 Event 20 Event.ENTER_FRAME 10, 14 Exécution, mode pour les
Concaténation 346 condenseWhite 434 Conditions 14, 18, 393 content 235 content LoaderInfo 107 Contenu alternatif 469 contextMenu 417, 419, 420 ContextMenuBuiltInItem 419 Contrôle du temps 59 Conversion degrés et radians 22 Convertir un fichier KMZ en DAE 303 Convertir un objet 64 en MovieClip 64 copyChannel 385 Couche, extraire 385 Couleurs	Déplacer un symbole (startDrag) 406 Détecter fin d'une animation 35, 47, 64 image du scénario 376 version du lecteur Flash 469 Développement back-office 134 front-office 134 Dictionnaires 484 Différence (opérateur !=) 267 Dimensions d'une page web 139 de la fenêtre de navigateur 437 displayState 442 Document, organiser 8 Données, gérer avec XML 121	Enveloppe vidéo 173 Ergonomie 48 Erreur chargement 131 exécution 131 Espace mémoire optimiser 14 typage 20 Étendre les propriétés d'un objet 64 Étiquette 423 créer 426 Event 20 Event.ENTER_FRAME 10, 14 Exécution, mode pour les squelettes 92

F	ForwardButton 190	gotoAndStop() 59
	FullScreenButton 190	Graphisme 331
F4V 162	MutteButton 190	Greensock 41
Quick Time 184	PauseButton 190	Greensock TweenMax 73
Voir aussi Vidéo	PlayButton 190	
filters 65, 264, 345	PlayPauseButton 190	
GlowFilter 268	Preview 174	Н
Filtres 65, 264, 345	scaleMode 273	
animation et interpolation 352	SeekBar 191	H-264, codec 179
appliquer 343	skin 172	niveaux 182
biseau 70	source 170, 273	hideBuiltInItem() 419
Bleu 348	stop 273	Hiérarchie, cinématique 77
correction des couleurs 343	Stopbutton 191	Historique de navigation dan
Cyan 348	VolumeBar 191	Flash 428
décliner 71	FLVPlayBackCaptioning 211	HTML 134, 465
flou 68, 340	Focus 413	importer dans Flash 430
halo 69, 70, 268	Fonction	lien vers Flash 423
Jaune 348	interrompre 274	tester les liens 427
Magenta 348	neutre 60	voir aussi Intégration
Mélangeur de couleurs 348	nommer 10	XHTML stricte
négatif 347	paramètre 60	htmlText 434
ombre portée 68, 69	Fonctionnalité applicative 48	Hyperlien 421
par lot 351	for 127, 131, 258	
propriétés 73, 75	for each in 259	
Rouge 347	Formats vidéo 140	I
saturation 345	Forme Voir Animation	
Vert 347		i, variable d'incrémentation
finally 133	FreeCamera3D 311	111, 118
fl 32	Fréquence 183	ik
Flash	FTP 168	getArmatureByName() 82
structurer un document 464	Full Flash 437	getBoneByName() 84
vs. HTML 466		getChildAt() 83
	G	headJoint 77
FlashPaper 451	d	IKArmature 78, 82
Flottement, effet 19	Galerie	IKBone 77, 82
FLV 162	animer 29	IKEvent 82
FLVPlayBack 169, 174, 187	d'images 99	IKJoint 82, 83
ActionScript 2 et 3 170	voir Image	IKManager 78, 82, 92
autoPlay 273	vidéo 3D 268	IKMover 82, 84, 94
BackButton 190	getChildAt 237, 258, 286	JKMover 78
boutons séparés (voir Vidéo)	getPixel 359, 360	tailJoint 77
189	gKaster 177	Image
BufferingBar 190	GlowFilter 65, 70	aléatoire 99
CaptionButton 190	, and the second	carroussel 251
contentPath 170	Google 464	externalisée 108
fenêtre 169	Google Sketchup 300	externe 99, 103, 107

galerie 99 XML 113 mur 3D 260 réaliser une texture raccord 24	IO_ERROR 132 IOErrorEvent 132, 134 ips Voir Cadence Itération 258	Liste d'affichage 82 modifier l'ordre 275 Livres 483 load 102
tremblante, éviter 27	itti ation 230	load() 118
Image de scénario Voir Scénario		Loader 100
iMovie 155	J	localConnexion 458
import 32	Jauge de chargement 103	
Importer	JavaScript 134	N.A
classes ActionScript 290 CSS dans Flash 430	Jouer le scénario 59	M
HTML dans Flash 430	Jouet le scenario 37	mask 340
PHP/MySQL 134 police 454	K	Masque conflit avec les polices de
un squelette 94 variable 434	KEY_DOWN 320	caractère 454
Imprimer un SWF 353	KEY_UP 320	dynamique 340 progressif (bords flous) 340
Incrémentation 26, 53, 59, 111	keyCode 320	Masquer
InDesign 450	•	pointeur Main au survol d'un
réaliser un livre au format SWF 250	L	bouton 35 symbole 37
Indexation 464	Label <i>Voir</i> Étiquette	Math 22
Initialisation 73	Langage, nommer les calques	ceil 231
int 20, 28	automatiquement 8	random 356
Intégration XHTML stricte 463	Lecteur Flash	Math.random 103
Interactivité 315	menu contextuel 417	Math.random() 101
objets dynamiques 122	pénétration (statistiques) 287	Maya 299
résoudre les conflits d'objets	Lecteur vidéo	Mémoire, optimiser 9, 14
imbriqués 400 Interface élastique flottante 443	H-264 pour Flash 6 et suivants	Menu
Internet Explorer, blocage des	191	contextuel 417
contrôles active-X 469	personnalisé 188	déroulant 407
Interpolation 34	Lévitation 18	MENU_ITEM_SELECT 420
durée 266	Lien	metaDataEvent 226
interrompre 36	HTML 421	Métadonnées 467
Tween 36	tester 427	créer dynamiquement 468
TweenMax 45	vers Flash 423	Microsoft Aero 279
Interrompre	Lipsync 86	Microsoft Window Movie
fonction 274	Lissage 250	Maker, exporter en AVI-DV 154
interpolation 413	Lisser	Mise en cache vidéo 139
un contenu chargé 235	graphismes vectoriels 335	Mode plein écran 437
Inverse kinematic <i>Voir</i> Cinématique inverse	images 331 vidéo 268	Modéliser 299
	.1000 =00	

MouseEvent 37	Occurrences, nommer 10	zone d'affichage de l'espace 3D
MOUSE_OVER 37	On2 VP6 137, 162, 163	309
MOUSE_UP 55	onCompleteListener 47	zoom 311
mouseX 21	onStartListener 47	parseCSS 433
Mouvement	Opacité 396	Particules 141, 149
caméra 3D 304	accessibilité 481	Pas
organique 86	opérateur != 267	d'accélération 12
progressif 18	Optimisation mémoire 9	d'incrémentation 26
MovieClip 64	Organiser Organiser	Passes d'encodage Voir Vidéo
afficher la main au survol 414	le code 9, 16	PDF 450
conflit d'interactivité 400	les fichiers du site 102	Performance d'affichage 441
convertir 64	overwrite 413	perspectiveProjection
fonction Bouton 389	over write 413	filedOfView 284
neutraliser, désactiver 414		focalLength 284
	Р	projectionCenter 283, 284
N		Photoshop 24
14	pageRanking 466	pitch() 314
name, propriété 83	Pages satellites 476	Pixelisation 250
navigateToURL 421	Panoramique 7	Pixellisation Voir Lisser les
Navigation, repères 218	défilement en boucle 23	images
NetStream 195	réaliser une texture raccord 24	Placer un objet au premier-plan
Netteté 27	sens de défilement 25	275
images tremblantes 27	PaperVision 289	playheadTime 204
voir aussi Lisser	différence entre	Plein écran Voir Affichage
new Tween() 34	Camera3D et	Podcast 168
nextFrame 377	FreeCamera3D 311	Poids standard d'une page web
Nodes 79	scène 3D et zone d'affichage	103
Nombre d'images par seconde 10	309	Point(X,Y) 84
Nommer	exemple en ligne 314	Pointeur, aspect du pointeur au
fichiers et dossiers 465	importer MovieClip dans l'espace 3D	survol 35
fonction 10	309	Points de repère 211
occurrences 10	objet 3D DAE 308	voir aussi Vidéo 217
Nuancier 357	installer 289	Police
Number 9, 16, 20, 28	intégrer 297	conflit avec les masques 454
numChildren 258	nativement 297	importer 451, 454
numemuren 238	ponctuellement 298	vectoriser 451
	nombre de polygones 300	Position
0	pitch 314	calcul 341
	placer une scène 3D entre deux	d'un clip 12
Object 463	MovieClip 309	relative 412
Objet	primitives 320	prevFrame 377
opacité 12	roll 314	PrintJob 353
rotation 12	yaw 310, 314	Progression du chargement 105,
visibilité 12	Z (propriété) 311	107

Propriétés 12	vidéo 377	Séparer 451
alpha 47	zone	setMetadata 468
animer 7, 14, 34, 44	cadre 363	Sites web et forums 484
courantes 14	fenêtre 363, 365	Sketchup 304
rotation 45, 46	jaillissante 363, 367	Skin 188
scaleX, scaleY 47	removeEventListener 14, 274	smoothing 250, 333
visible 37	Répartir vers les calques 8	Son, accessibilité 481
y 46, 52	RESIZE 446	Sorenson Spark 137, 162, 163
Pseudo extension 10	Résolution 139, 437	Sous-classe 64
Puits de couleurs 357	limite de Flash 349	Sous-titrage 211
push() 384	Ressources 483	O
	Restauration 73	Spamdexing (spam) 465
0	Retard 18, 20	Squelette 77
Q	Retarder une action 59	animer 84
Qualité 250	Richmédia 48, 239	avec formes graphiques 86
Qualité d'affichage <i>Voir</i> Lisser	roll () 314	charger dans un nouveau SWF 94
les images	RollOver vidéo 276	construire 80
Quick Time 184	root (racine) 237	contrainte de mouvement et de
C	rotation, propriété 12	rotation 80
5	RVBA, extraire une couche 385	définir 81
R		humain 82
Racine (root) 237	S	mode Exécution 80
	3	nom d'occurrence 80
Ralenti 18 effet 21	scaleMode 445	ordre d'affichage des liaisons
Rastérisation 250	scaleX, propriété 12	81
	scaleY, propriété 12	programmer 77
Ratio des pixels 138	Scénario 7	stage 237, 445
Rebondissement 18	atteindre l'image	stageDisplayState 443
Rectangle 54	précédente 377	stageHeight 264, 445
Redimensionner la fenêtre 446	suivante 377	stageWidth 264, 445
Référencement 463, 464	compter les images 245	startDrag 406
Relief 361	détecter l'image active 376	startDrag, stopDrag 54
anaglyphe, créer	Scénariser 48	Statistiques, pénétration du
à partir de 2 images 370 ActionScript 375	Scène	lecteur Flash 287
Photoshop 363	hauteur 264	Streaming 210
couches RVB 365	largeur 264	String() 434
interfaçage dynamique 377	mode d'affichage 445	Structure
lunettes	propriété 3D 284	conditionnelle 28
actives 361	SCPlugin 292	d'un document Flash 235
passives 361	Scroll 54	flottante 476
réseau lenticulaire 361	seek 204, 207	Styles 214, 215
technique de prise de vue 362	seekToNavCuePoints 218	CSS 473

Suivre le pointeur 18	Timer 59, 93, 286	durée 266
Supprimer	détecter la fin 64	enchaîner des interpolations à
écouteur 113	stopper 64	47
symbole 451	Tortoise SVN 290	importer la classe gs 67
un SWF chargé 232	totalFrames 245	mise à jour 41
Surface utile 139, 437	Touche du clavier Voir Clavier	onCompleteListener 412
SVNX 292	trace 131	onStartListener 412
swapChildren 247	transitionManager 60, 62, 64	overwrite 413
SWFAddress 428	définition 62	propriétés 73
SWFObject 428	paramètres 62	TweenMax gs, importer 67
SWFObject2 469, 476	Transparence 396	Typage 14, 28
Swift 3D 299	accessibilité 481	nombre 20
Switch case 393	animer 13	Typographie 451
Symboles	du Flash 435	voir Police 454
déplacement 25	Transtypage 64	
exporter pour ActionScript 383	chaîne de caractères String()	U
nom d'occurrence 8, 14	434	
placer le centre 24, 247	Trucage vidéo 141, 152	UILoader 110, 112
position 27	try 133, 134	uint 20, 28
transparent 48	Tutoriels vidéos 483	unloadAndStop 235, 238
zone active 48	Twease 29	URL 421
Syntaxe chameau 14	Tween 29, 32, 38, 250	URLLoader 118, 119
séparateurs 10	enchaînement 35	URLRequest 118
Systèmes de navigation 389	instabilité 36	URLVariables 434
	interpolation 36	useHandCursor 35
т	interrompre 36	
Т	TweenEvent 32, 35, 38, 47	
Tableau 384	COMPLETE 47, 90	V
target 129, 131, 210, 392	enchaînements 36	¥7-1
Technologie 466	MOTION_CHANGE 36	Valeur aléatoire 356
Teinte aléatoire 355	MOTION_FINISH 35, 36	générer 101
	MOTION_LOOP 36	incrémenter 26
Temporiser une action dans le temps 59	MOTION_RESUME 36 MOTION_START 36	Variabilité du débit <i>Voir</i> Vidéo
Tête de lecture, boucle 420		Variables 18, 28, 33, 51
Text Layout Framework 458	MOTION_STOP 36 TweenLight 41	globale 116
Texte 464	TweenLite 29	importer dans Flash 434
accessibilité 481		locale 116
défilant 54	TweenMax 29, 44, 230, 250	si 59, 61
dynamique 431, 452	assistant 42 bézier 42	valeur par expression 35
statique 452	carte interactive 38	VBR, CBR Voir Vidéo
vectoriser 451	courbe de bézier 44	Vectorisation, accessibilité 481
Voir aussi Police	définition 45	Vectoriser 451
Timecode 208, 221, 226	delay 73	Vélocité 18, 53

Version lecteur Flash, pénétration (statistiques) 287	compression 140, 152, 166 configuration utilisateur 140,	remappage temporel 165 rembobiner 205
Vidéo 137, 177	141	rollOver 276
3D 268	débit 151, 165	source 208
accélérer 204	détecter la fin 217	sous-titrage 211
Adobe After Effects	détourage du fond vert 153	activer/désactiver 215
manuel d'apprentissage 153	dimensions 268	standard 139
Mode de fusion des calques	standard 164	streaming 210 timecode 221, 226
198	effet 3D 174 enchaîner les vidéos 217	trame 197
relief 377		entrelacement, progressif
Adobe F4V 162	encodage 217, 219 exporter pour Flash 6 191	138
Adobe Media Encoder	externaliser 139	transparence 137, 163
Autres 168	F4V 177, 179, 218, 224, 227	voir aussi Adobe After Effects,
FTP 168	Flash Media Server 210	Adoble Media Encoder,
Passes d'encodage 165	flux simultanés 139	Adobe Premiere Pro, Apple
Réglages avancés 166	FLV 162, 218, 224, 227-228	Final Cut Pro, Apple iMovie, Apple Motion, Cadence, On2
Réglages de débit	générateur de particules 141	VP6, Quick Time, Sorenson
CBR, VBR 165	HD 139	Spark, Window Movie Maker
Variabilité du débit 165	images-clés 166, 204, 220	volume audio 205, 206
affichage en relief 377	intégrée 229	videoEvent 215
agrandir 197	problème de	Viewport3D 308
Apple Motion 165	synchronisation 165	Visibilité 396
arrêter 204, 232, 235	intégrer 139	visible 396
dans un SWF imbriqué 191	interactive 201	propriété 12
depuis le document racine 234	lecture automatique 203 lire 204	volume() 205
Audio 183	avec des boutons	
via FLV 161, 167	préprogrammés 172 en arrière (vitesse variable et	W
boucle 215	fluide) 227	Window Movie Maker 154
bruit 139	lissage 268	wmode 435
cadence 138, 143	lumière (prise de vue) 152	windue 455
capture 139	métadonnées 226	
chapitrage 207, 211, 218	mise en cache 139, 203	X
codage différentiel 180	Points de repère (CuePoints)	*/// 10
codec 137	dynamiques (en	x, propriété 12
compatibilité 163	ActionScript) 226, 227 événement 217, 223	XFL 450
format vidéo de Flash 140	navigation 217, 220	XML 113, 211, 217
composant FLVPlayBack Voir	profondeur de couleur 143	atteindre attribut 119
FLVPlayBack	qualité et échelle 139	nœud 119
composite dans Flash 169	ratio des pixels 138, 143	cibler un attribut connu 120

créer un document XML 117 importer du HTML 119 length() 119 lire 119 mécanisme 116 nœud 117 optimiser la gestion des données 121 site dynamique 134 structure 116 **XMP** 467

Y

y, propriété 12 yaw 310

Z

z, propriété 279 Zone active sur un symbole 48 Zone de défilement 54 Zoom 122, 260

Placer un nom d'étiquette sur une image du scénario 426

Index des notes et encadrés

Aides au développement de

Accessibilité

3D

Accéder directement à une

469, 476, 481, 482

Réaliser une image de type motif, bibliothèque d'objets 3D KMZ contenus spécifiques pour les malentendants et les 300 raccord, pour panoramiques 24 malvoyants 481 Résolution limite des images 349 Aide en ligne de PaperVision 328 Informations sur l'accessibilité Système hexadécimal 356 Conditions d'utilisation de la dans Flash 481 banque d'objets 3D Google HTML Sketchup 300 Animation Créer un lien HTML sans Gestion des lumières dans Animation de filtres 353 ActionScript dans Flash 422 PaperVision 328 Définir les coordonnées d'une Les cibles d'affichage 421 Guide de référence PaperVision courbe de Bézier pour la classe Tester les liens HTML sur un 328 TweenMax 44 serveur distant 427 Définition d'une interpolation Identifier des groupes d'objets Langage TweenMax 45 pour PaperVision 303 3D avec la classe Caurina 250 Définition de la classe Interactivité sur les objets 3D 328 3D avec la classe Tween 250 transitionManager 62 La sous-classe Camera3D 311 3D avec la classe TweenMax 250 Éviter les images tremblantes 27 La vidéo en relief 377 Arrêter un écouteur 55 Gérer un ascenseur avec une Les primitives de PaperVision Arrêter un Timer 64 accélération 53 320 Arrondir une valeur 101 L'assistant TweenMax 42 Limite de taille des fichiers 3D Calcul de la durée d'un Mise à jour de la classe 300 chronomètre 59 GreenSock TweenMax 41 Lisser les images pour la 3D 250 Calculer avec la class Math 22 Stabiliser une interpolation Optimiser un site 3D 314 Calculer un pas d'incrémentation Tween 36 Placer un symbole par-dessus un 26 Conception et intégration obiet 3D 309 Ciblage dans un document Flash Dimensions standard d'une vidéo AS3 237 Réaliser un livre interactif avec pour le Web 164 InDesign 250 Ciblage des boutons entre SWF Graphisme imbriqués 415 Sketchup 304 Centrer les images avec le Composant FLVPlayBack pour Vérifier les valeurs de ses propres ActionScript 2 ou 3 170 touches clavier 323 composant UILoader 110 Définition des paramètres du Conversion de degrés en radians À retenir 14, 18, 22, 28, 38, 48, 55, filtre flou BlurFilter 68 et inversement 22 64, 75, 85, 91, 94, 97, 103, 107, Créer un fichier XML 117 Définitions des paramètres du 113, 121, 131, 134, 147, 153, 169, filtre biseau BevelFilter 70 Définir les chemins pour les 174, 184, 188, 191, 197, 199, 207, requêtes externes 102 Définitions des paramètres du 211, 215, 216, 227, 231, 238, 251, filtre halo GlowFilter 70 Définition du constructeur Point() 260, 268, 279, 287, 299, 304, 311, Définitions des paramètres du 84 314, 329, 335, 337, 343, 353, 355, filtre ombre portée Détecter la fin d'une boucle 356, 360, 377, 387, 395, 404, 407, DropShadowFilter 69 d'itération Timer 64 414, 420, 422, 427, 430, 434, 435, Faut-il renseigner toutes les Différence entre Loader et 443, 449, 451, 458, 462, 464, 467,

propriétés pour les filtres ? 73

URLLoader 119

Différence entre target et	Target et currentTarget 415	Vidéo
currentTarget 129	Typage des nombres 20	Chapitrage vidéo avec les points
Doit-on toujours placer un	Logiciel	de repère 211
écouteur sur un chargeur pour activer l'affichage d'un contenu chargé dynamiquement ? 103	Centraliser les classes ajoutées avec les chemins de classe	Comment réactiver les champs de dimensionnement de
Enchaîner des actions à une	(classPath) 42	l'encodeur ? 220
interpolation TweenMax 47	Installer une classe importée 428 Localiser les classes natives de	Composant FLVPlayBack pour ActionScript 2 ou 3 170
Exemples de ciblage de contenus dans des SWF imbriqués (AS2 et AS3) 237	Flash 32 Navigation Différence entre les propriétés	Conflit entre la cadence des vidéos et la cadence des animations Flash 165
Exporter un symbole pour ActionScript 383	alpha et visible 396 Menus avec des symboles	Définition des composants associés à FLVPlayBack 190
Générer une valeur aléatoire 101	boutons 395	Désactiver et réactiver l'affichage
L'ordre numérique en ActionScript 112	PDF Convertir un PDF en SWF avec	des sous-titres 215
Le transtypage 64	FlashPaper 451	Enchaîner plusieurs vidéos à la suite 217
Les classes 11	Création dynamique de fichiers	Encoder en FLV avec Apple
Les noms du fichier appelant 95	PDF 451	Compressor 147
Les propriétés en ActionScript 3	Différence structurelle entre Flash et Acrobat 450	Étendre les formatages du document XML pour les sous-
Les touches de commande 321	Sites Full Flash	titres 215
Lire le XML 119	Activer l'accélération matérielle pour les contenus riches 441	Exporter le projet After Effects natif directement vers Flash 153
Mécanisme d'un tableau (Array)	Désactivation des contrôles du	Flash Media Server 210
384 Mécanisme d'une boucle for 127	clavier en plein écran 441 Modes d'affichage de la scène	Icône de sélection de fichier du
Nommer les occurrences 10	445	composant FLVPlayBack 171
Nommer une fonction 10	Principe de flottement façon	Intégrer physiquement une vidéo dans le scénario 229
Options d'impression 354	styles CSS 449	Le composant CaptionButton
Où placer l'écouteur et la fonction ? 63	Quitter le mode Plein écran avec Echap 443	215
Paramétrer un composant <i>via</i>	Squelette	Les formats pris en charge par Adobe Media Encoder 156
ActionScript 174	Cibler un segment 84	Les images-clés 166
Philosophie de AS3 9	Construire un squelette pour la programmation 80	Mécanisme du codec On2 VP6
Pourquoi transtyper un chargeur en MovieClip ? 235	Création de squelettes humains	162 Modifier la cadence des images
Propriété overwrite pour TweenMax 413	Heure de création et exécution 78	165
Propriétés d'alignement de la	Typographie	Paramètre source des composants vidéos 172
scène 446	Affichage des polices dans Flash 452	Reconstituer un univers 3D avec
Propriétés 3D de la scène 284	Le nouveau moteur de texte Text	une vidéo aplatie 174
Que faire des classes après	Layout Framework 458	Trucage avec captation sur fond
compilation des SWF ? 42	Les polices et les masques 454	vert 152
Syntaxe chameau et séparateurs 10	Vectoriser un texte dans Flash 451	Utiliser le FLV pour l'audio, la vidéo ou les deux 161

Le Campus

ActionScript 3 et le motion design

Programmer facilement la vidéo en HD, la 3D, le relief, les Tweens... dans le scénario!

Vous êtes graphiste et débutez en ActionScript 3 ou utilisez déjà ActionScript 2. Vous aimeriez contrôler les contenus et profiter des performances de ce nouveau langage directement dans le scénario de Flash. Ce livre est fait pour vous!

Vous y trouverez des solutions nouvelles, concrètes et clés en main. Vous y acquerrez les notions d'ActionScript indispensables pour pouvoir concevoir, sans l'aide d'un développeur, des interfaces sophistiquées et dynamiques à partir de symboles placés dans le scénario.

Sont abordés de façon claire et didactique : la vidéo en qualité HD avec toute l'interactivité, la 3D native, l'animation, les classes d'animation Tweens et TweenMax, les effets et les filtres, la gestion de documents imbriqués désormais plus complexe. Mais aussi de nombreux concepts inédits, comme le référencement de documents Flash, le relief, l'animation de squelettes et la 3D avec PaperVision.

Plus qu'un livre d'apprentissage, cet ouvrage rassemble toutes les techniques qu'un bon web designer doit aujourd'hui connaître. Un outil indispensable en production.

À propos de l'auteur

Arzhur Caouissin, auteur et réalisateur multimédia indépendant, est formateur depuis plus de dix ans à l'école des Gobelins, chez Pyramid et à l'INA. Spécialisé dans la technologie Flash, il est apprécié pour sa vision transversale et artistique des différents médias, et pour son expertise en création de contenus web enrichis.

Codes sources et galerie sur www.pearson.fr!

PEARSON Pearson Education France 47 bis rue des Vinaigriers 75010 Paris Tél.: 01 72 74 90 00

Fax: 01 42 05 22 17 www.pearson.fr

Table des matières

- Les animations en ActionScript
- Interpolations et interactivité avec les classes Tween et TweenMax
- · Les transitions d'effets et de
- La programmation de squelettes
- · Les galeries d'images
- La vidéo standard et composite en FLV
- La vidéo HD en F4V
- La vidéo interactive
- La 3D native
- · La 3D et PaperVision
- · API d'affichage et de colorimétrie
- · Le Web en vrai relief
- · Les systèmes de navigation
- La communication Flash/HTML
- · La gestion de sites Full Flash
- Les solutions de référencement du Flash
- L'accessibilité dans Flash
- Ressources

Niveau: Intermédiaire / Avancé

Catégorie: Graphisme web /

Programmation

Configuration: Flash CS4 et ultérieur,

Mac OS / Windows

ISBN: 978-2-7440-4128-0

