

Différence entre target et currentTarget

Dans l'expression `evt.target.name`, nous désignons la propriété `name` (le nom) de l'élément activé contenu dans l'objet (`target`) auquel est rattaché la classe invoquée par l'écouteur, et donc, l'événement qui lui est associé (`evt`). Cette expression (`target`), est employée lorsque plusieurs objets isolés dans un même et unique conteneur exécutent la même fonction.

Dans l'expression `evt.currentTarget.name`, nous désignons la propriété `name` (le nom) de l'objet attaché à l'écouteur lui-même (`evt.currentTarget`). Cette expression (`currentTarget`) est utilisée lorsque plusieurs objets isolés et sans relation exécutent la même fonction.

Considérons l'exemple suivant : prenons deux ensembles, un ensemble de lettres et un ensemble de chiffres. Plaçons un écouteur sur chacun d'eux. Les deux ensembles invoquant la fonction alors (`evt : Event`), la propriété `target` renverra la lettre ou le chiffre qui aura intercepté le clic, tandis que `currentTarget` renverra le groupe auquel l'écouteur est appliqué et dont l'événement a été intercepté.

Ainsi, il n'est pas nécessaire de connaître à l'avance l'objet qui va être activé et qui va exécuter la fonction pour lui affecter un comportement.

Dans cette fonction, nous remplaçons l'action `trace`, citée en exemple, par les instructions suivantes :

```
cibleVignettes_mc.carreBlanc_mc.visible=true;
titre_txt.htmlText=donneesXML.photo[evt.target.name].titre.toString();
legende_txt.htmlText=donneesXML.photo[evt.target.name].legende.toString();
cheminPhoto=new
↳ URLRequest(donneesXML.photo[evt.target.name].image.toString());
chargeurPhoto.load(cheminPhoto);
TweenMax.to(cibleVignettes_mc.carreBlanc_mc, 1,
↳ {x:(ecartEntreVignettes*(evt.target.name-1))-2, ease:Strong.easeOut});
cible_mc.scaleX=0.1;
cible_mc.scaleY=0.1;
cible_mc.x=mouseX;
cible_mc.y=mouseY;
TweenMax.to(cible_mc, 3, {x:100, y:50, scaleX:1, scaleY:1,
↳ ease:Strong.easeOut}); // transition Zoom
```

À chaque clic sur une vignette, nous réactivons l'affichage du carré blanc, masqué par défaut, avec la propriété `visible` passée sur `true` :

```
cibleVignettes_mc.carreBlanc_mc.visible=true;
```

Puis, nous modifions le texte contenu dans les champs de texte dynamiques `titre_txt` et `legende_txt` :

```
titre_txt.htmlText=donneesXML.photo[evt.target.name].titre.toString();
```

Pour déterminer le nœud actif, nous utilisons un nœud de l'arbre XML `donneesXML` (`donneesXML.photo`) et spécifions le nom de l'objet cliqué avec `evt.target.name`, entre crochets.

Nous procédons de même pour désigner le nœud correspondant à l'URL de l'image agrandie :

```
legende_txt.htmlText=donneesXML.photo[evt.target.name].legende.toString();
```

Le chargeur relance ensuite l'affichage de l'image agrandie :

```
chargeurPhoto.load(cheminPhoto);
```

Enfin, nous appliquons une transition TweenMax sur la position horizontale du carré blanc pour créer ainsi l'effet de déplacement. Seul le paramètre x doit être calculé et défini. Il suffit de récupérer au travers de son nom l'index de position de la vignette et de le multiplier par l'espacement existant entre chaque vignette (90 pixels) puis de retrancher le décalage de 2 pixels nécessaire pour assurer la marge du carré blanc.

Attention, comme lors du chargement la première vignette est déjà en place, il nous faut en tenir compte et démarrer l'incréméntation à 1, donc retrancher 1 à la valeur d'index de la vignette. À défaut de cette correction, nous observerions un décalage de 90 pixels vers la droite.

Nous terminons enfin le code par la définition d'actions sur les boutons suivant et retour, pour modifier la position du conteneur cibleVignettes_mc, à chaque clic sur les flèches correspondantes :

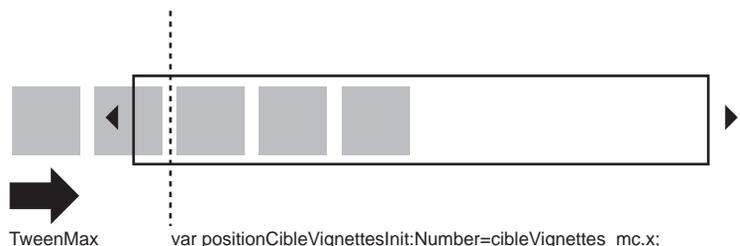
```
//----- Boutons NAVIGATION

retour_btn.addEventListener(MouseEvent.CLICK, retourDOWN);
function retourDOWN(evt:MouseEvent) {
    if (cibleVignettes_mc.x<=positionCibleVignettesInit*2) {
        TweenMax.to(cibleVignettes_mc, 1,
            {x:cibleVignettes_mc.x+ecratEntreVignettes, ease:Elastic.easeOut});
    }
}
suivant_btn.addEventListener(MouseEvent.CLICK, suivant);
function suivant(evt:MouseEvent) {
    if (cibleVignettes_mc.x>=0) {
        TweenMax.to(cibleVignettes_mc, 1, {x:cibleVignettes_mc.x-ecratEntreVignettes,
            ease:Elastic.easeOut});
    }
}
}
```

Dans chacune de ces deux fonctions, l'animation TweenMax est activée uniquement si la position de la cible des vignettes l'y autorise. Plus exactement, l'animation est lancée si, pour le déplacement du menu dans le sens du retour, sa position courante reste inférieure à sa position initiale (voir Figure 5.16). Nous multiplions cependant la valeur par deux afin de recentrer arbitrairement l'objet dans la scène. Mais ce n'est pas une obligation.

Figure 5.16

Schéma du repositionnement du conteneur de vignettes.



Nous procédons au même type de calcul dans l'autre sens, avec la deuxième fonction. Nous spécifions alors que l'animation TweenMax peut se produire tant que la position de l'objet ne passe pas par la limite de gauche qui vaut 0.

À retenir

- Pour gérer une construction dynamique d'interface dont les mécanismes de mise en forme se répètent, nous pouvons utiliser une boucle for.
- Pour associer une action à un objet généré dynamiquement, il est possible de cibler l'objet en utilisant sa propriété name.
- La propriété target se distingue de la propriété currentTarget en cela qu'elle cible l'objet activé, contenu dans l'objet attaché à l'écouteur, alors que currentTarget cible uniquement l'objet attaché à l'écouteur. La propriété currentTarget est donc employée pour distinguer plusieurs objets isolés qui exécutent la même fonction. La propriété target est utilisée pour désigner plusieurs objets d'un même conteneur qui exécutent la même fonction.

Résoudre les erreurs éventuelles au chargement et à l'exécution

Des erreurs de chargement ou d'exécution peuvent apparaître lorsque vous produisez avec des fichiers externalisés (pour le chargement) ou lorsque vos développements sont lourds ou exécutés sur des systèmes instables ou à bout de souffle (pour l'exécution). Une erreur, fut-elle bénigne, peut, dans certains cas, neutraliser l'ensemble de votre application si celle-ci n'est pas orientée vers une action alternative.

Dans cette section, nous présentons les instructions qui permettent d'identifier les erreurs et celles qui permettent de les résoudre, à travers des actions trace.



Exemples > ch5_galleriesImages_6.fla

Dans la scène principale du document "ch5_galleriesImages_6.fla", au-dessus du calque fond_mc, figure un unique calque actions (voir Figure 5.17).

Figure 5.17

Aperçu de la fenêtre de scénario de la scène principale.



Dans le calque nommé actions, nous pouvons lire le code suivant :

```
//----- Chargement du XML
var chargeurXML:URLLoader = new URLLoader();
```

```
//chargeurXML.load(new URLRequest("xml/galerie.xml"));
chargeurXML.load(new URLRequest("xml/g.xml"));
chargeurXML.addEventListener(Event.COMPLETE, XMLComplet);

chargeurXML.addEventListener(IOErrorEvent.IO_ERROR, siErreur);
function siErreur(evt:IOErrorEvent) {
    trace("Erreur de chargement");
}

function XMLComplet(evt:Event) {
    trace("Document XML chargé");
    //
    try {
        trace("Document correctement exécuté");
    }
    catch (monErreur:Error) {
        trace("Document mal exécuté. Voici l'erreur = "+monErreur);
    }
    finally {
        trace("Instruction post analyse. Exécutée avec ou sans erreur, après l'analyse.");
    }
}
```

Dans notre document, nous chargeons le fichier XML utilisé pour la réalisation de la galerie photo. Mais aucune action n'est distribuée pour traiter les données. Nous avons simplement placé dans le code les instructions qui permettent, à chaque étape, de localiser tout éventuel problème de chargement ou d'exécution. Les actions trace qui apparaissent permettent alors d'identifier l'apparition de tel ou tel problème et, si besoin, de substituer ces actions trace par des instructions qui proposent un contenu alternatif à ce qui devait apparaître en l'absence d'erreur.

Ainsi, dans le code de ce document, nous avons mis en commentaire la ligne qui fait référence à l'URL du document XML valide. Puis, nous l'avons dupliquée en remplaçant le nom du fichier valide par un nom invalide. Ainsi, le fichier qui n'existe pas génèrera une erreur de chargement.

Pour traiter une erreur liée au chargement, nous plaçons un écouteur qui utilise l'événement `IO_ERROR`, disponible *via* la classe `IOErrorEvent`. Cela donne :

```
chargeurXML.addEventListener(IOErrorEvent.IO_ERROR, siErreur);
function siErreur(evt:IOErrorEvent) {
    trace("Erreur de chargement générée volontairement dans le cadre de cet exemple");
}
```

Lorsque le document est publié, le fichier inexistant n'est pas trouvé. Le message, inscrit dans l'action trace ("erreur de chargement générée volontairement dans le cadre de cet exemple"), apparaît aussitôt dans la fenêtre de sortie. En réactivant l'instruction `URLRequest` valide et en neutralisant la commande erronée, le message lié à l'erreur de chargement n'apparaît plus lorsque le document est à nouveau publié.

Plus loin, une fois le document XML chargé correctement, si l'instruction valide a été réactivée, nous pouvons tester les autres formes d'erreurs liées à l'exécution. Les commandes

qui gèrent ce type d'erreurs sont placées à l'intérieur de la fonction appelée à l'issue du chargement du document XML.

Dans un premier temps, nous vérifions que le document a effectivement bien été chargé. Pour cela, nous introduisons une simple action trace au début de la fonction. Cette action ne vérifie rien sinon qu'elle ne peut être exécutée que si le document a effectivement bien été chargé et elle nous informe donc sur le succès du chargement :

```
trace("Document XML chargé");
```

Ensuite, pour contrôler la validité des instructions qui seront exécutées après le chargement, toutes les lignes de code de cette fonction doivent être introduites dans le bloc d'instructions `try{}`. À défaut, les actions non incluses entre les deux accolades qui suivent la méthode `try` ne seront pas vérifiées. Pour tester la validité des actions contenues dans le bloc d'instructions `try{}`, nous y plaçons une action trace qui confirme la bonne exécution de l'ensemble du programme :

```
try {
    trace("Document correctement exécuté");
}
```

Un bloc `try` doit toujours être suivi, soit d'un bloc `catch{}`, soit d'un bloc `finally{}`, soit des deux.

Le bloc `catch{}` propose l'exécution d'une instruction alternative dans le cas où une erreur aurait été détectée. Par exemple, vous pouvez afficher à travers le bloc d'instruction `catch{}` un message. Ce message peut signaler, à l'utilisateur, une erreur d'exécution indépendante du site et lui suggérer de recharger le document ou de se reconnecter ultérieurement. Vous pouvez aussi exécuter une version allégée de votre programme proposé initialement dans le bloc `try{}`, mais cela augmente naturellement la charge de votre développement. Dans notre exemple, le bloc `catch{}` exécute une action trace qui affiche le type d'erreur rencontré, en invoquant la classe `Error` relative à cette commande :

```
catch (monErreur>Error) {
    trace("Document mal exécuté. Voici l'erreur = "+monErreur);
}
```

Le bloc `finally{}` s'exécute, lui, quel que soit le résultat obtenu par les blocs `try{}` et `catch{}`. Il sert à ponctuer l'exécution d'un programme par une action complémentaire. Ce bloc est moins usité que les deux précédents, mais comme son nom l'indique, il permet de finaliser un programme et orienter éventuellement l'utilisateur sur les actions à conduire. Il peut aussi servir à réinitialiser les valeurs initialisées lors de la tentative de la première méthode `try{}`. Dans notre exemple, une action trace termine le programme :

```
finally {
    trace("Instruction post analyse. Exécutée si erreur ou sans erreur, après l'analyse.");
}
```

Mais, étant exécutée même lorsqu'il n'y a pas d'erreur, il convient d'employer le bloc `finally{}`, avec une structure conditionnelle qui initialise les valeurs, seulement si nécessaire.

À retenir

- Il est possible de gérer les erreurs de chargement grâce à la classe `IOErrorEvent`.
- Il est possible de gérer les erreurs d'exécution grâce à la méthode `try` et d'introduire une action alternative avec la méthode `catch`.
- Une méthode `finally` permet de ponctuer, à travers une structure conditionnelle, la détection d'une erreur par une instruction complémentaire, si nécessaire.

Gestion de site dynamique avec XML

Il se distingue deux types de développement : le développement front-office et le développement back-office.

Le développement front-office désigne la programmation effectuée sur un ordinateur local et exécutée dans le navigateur côté client. Même si ActionScript permet aussi de réaliser des développements côté serveur, ActionScript, JavaScript, HTML, sont des langages de développement utilisés généralement en front-office.

Le développement back-office consiste à programmer des applications qui sont exécutées côté serveur, et non plus localement. Si vous les réalisez localement, vous devez installer sur votre poste de travail un serveur d'émulation pour tester l'exécution de vos programmes. PHP, par exemple, est un langage de scripts très largement utilisé et exécuté côté serveur. Ce langage est en mesure de traiter des données depuis une base MySQL placée sur un serveur distant, et donc, de centraliser des données en vue d'être redistribuées dans un site. PHP peut également convertir assez simplement, les données stockées dans une base MySQL, en flux XML, comme si vous disposiez finalement d'un fichier XML localisé et stocké côté client.

Dès lors que vous savez exploiter XML dans la construction d'interfaces dynamiques en Flash, vous savez donc aussi appeler un même flux XML généré depuis un serveur distant, avec une application réalisée dans un langage dynamique tel que PHP, par un prestataire développeur back-office tiers, par exemple. Vous pouvez alors accéder, indirectement, et grâce aux instructions PHP, à des données centralisées dans une base MySQL, sous la forme d'une simple requête XML ou d'un simple lien vers un fichier PHP, que vous passez alors en paramètre de la méthode `URLRequest()` à l'intérieur de votre Flash (par exemple : `var chemin:URLRequest= new URLRequest("http://www.monsite.com/scriptQui-RenvoieUnFluxXML.php");`). Il devient donc relativement simple de confier la gestion des données à un développeur back-office et de vous concentrer sur votre propre développement en local en ActionScript, sans avoir à coder une seule ligne en PHP, sans jamais avoir à échanger le moindre fichier avec votre prestataire, ni installer quelque serveur que ce soit sur votre poste de travail, ni même placer vos documents Flash sur un serveur d'exécution pour les publier.

Il existe une autre alternative à la relation client/serveur, à base de Flash. C'est AMFPHP, qui permet à Flash de directement communiquer avec un serveur et inversement.

Cette méthode de répartition des tâches entre développeur front-office et back-office, grâce au XML, s'avère particulièrement efficace en production, car elle permet au graphiste codeur, de lui épargner la gestion d'un serveur ou d'un émulateur de serveur et au développeur back-office, de s'abstraire d'un document Flash parfois complexe à dépecer. Cette méthode permet, aussi, de bien séparer les tâches en programmation et de garantir que chaque partie concernée puisse se concentrer sur sa spécialité, sans jamais avoir à soumettre quelque partie que ce soit de son travail à un autre, tout en reliant pourtant dynamiquement les fichiers entre eux.

Synthèse

Dans ce chapitre, vous avez appris à optimiser considérablement le poids de vos documents lorsqu'ils affichent des images en grand format, en externalisant les contenus dans des répertoires placés au sein de votre site. Vous avez vu comment réaliser des galeries d'images et interagir avec des objets placés dynamiquement. Vous avez appris à centraliser des informations dans un document XML pour simplifier la maintenance de votre site. Enfin, vous savez maintenant contourner les erreurs éventuelles, observées au chargement ou à l'exécution des programmes. Vous êtes à présent en mesure de créer des interfaces dynamiques simples, et d'externaliser vos données dans un fichier XML. Vous êtes également au fait de la manière de traiter des données générées dynamiquement depuis une base placée sur un serveur distant si celles-ci sont distribuées sous la forme d'un lien ou d'une requête XML.

6

La vidéo standard et composite en FLV

Introduction

La gestion de la vidéo au sein de Flash est assez proche de celle de tout autre type de contenu. De cette manière, nous pouvons envisager son intégration indépendamment de son cadre habituel de diffusion, avec des dimensions libres et de l'interactivité, comme si nous traitons une image en somme.

Flash intègre la gestion de contenus vidéos depuis la version 6 (MX). Mais il compte aujourd'hui trois formats que nous regroupons dans deux grandes familles : le format FLV, flexible et universel, et le format F4V, plus "classieux" et récent. Le premier intègre deux codecs, le Sorenson Spark et le On2 VP6 tandis que le second propose uniquement le H-264. Nous abordons les spécificité du H-264 à travers le format F4V dans le chapitre suivant. Dans ce chapitre, nous détaillons l'encodage de la vidéo au format FLV avec ces deux premiers codecs, plus anciens.

Dans sa version 6, le lecteur Flash utilisait le codec Sorenson Spark, très compact, et a ainsi largement contribué à la diffusion à grande échelle de la vidéo sur le Web.

Avec Flash 8, l'apparition du codec On2 VP6 a permis d'améliorer la qualité des vidéos mouvementées dans Flash et de gérer la couche alpha éventuellement intégrée au flux vidéo.

Par vidéo standard, nous entendons donc l'ensemble des vidéos publiées au format FLV, encodée en Sorenson Spark, sans couche transparente, destiné à un affichage simple et compatible avec les plus anciennes versions du lecteur Flash (version 6).

Par vidéo composite, nous entendons la composition d'interfaces à partir de sources vidéos qui possèdent une couche transparente (couche alpha), encodée en on2VP6 avec la couche alpha. La transparence, lorsqu'elle est activée, permet de réaliser des interfaces graphiquement élaborées telles que l'incrustation d'un personnage filmé sur fond vert et détourné, des animations de particules, la superposition d'éléments de décor animés, des espaces publicitaires en premier plan d'une mise en page, etc.

Dans ce chapitre, nous détaillons d'abord les contraintes générales liées à la gestion d'une vidéo pour le Web. Nous abordons ensuite les techniques élémentaires de publication d'une vidéo depuis un logiciel tel que Apple Motion, disponible dans la suite Apple Final Cut Studio, ou Adobe After Effects de la suite vidéo Adobe, avec de la transparence. Nous voyons aussi l'encodage de vidéos standard vers Flash pour un export au format FLV, depuis tout type de logiciel. Enfin, nous traitons l'intégration simple de ces vidéos dans Flash, à l'aide du composant FLVPlayback.

À l'issue de ce chapitre sur la vidéo standard et composite au format FLV, vous serez en mesure de réaliser des interfaces contenant des vidéos simples, sophistiquées et optimisées pour toutes les versions de Flash qui acceptent la vidéo.

Concevoir la vidéo pour le Web

Lorsque vous préparez un contenu vidéo à destination du Web, vous devez considérer que sa diffusion doit obéir à quelques caractéristiques liées à la nature même du support de diffusion. Voici quelques questions/réponses qui vous guideront sur la création de contenus vidéos à destination du Web, et plus particulièrement, vers Flash.

Quel ratio de pixels ? Une vidéo échantillonnée pour le Web doit être définie en pixels carrés, de ratio 1,00. Les écrans d'ordinateur (qui exécutent la page web) possèdent une trame de pixels carrés, ce qui les distingue des écrans de télévision dont le ratio de pixels est respectivement 1,09 en PAL 4/3, 1,46 en PAL grand écran, 0,9 en NTSC 4/3 et 1,21 en NTSC grand écran.

Il est possible de convertir le ratio du format des vidéos à partir des logiciels Motion et After Effects. Pour connaître le ratio de vos vidéos, ouvrez-les dans l'une de ces applications et observez leurs propriétés. Pour adapter le document au format pixels carrés, définissez, dans les boîtes de dialogue qui apparaissent lors de la création d'une nouvelle séquence vidéo, un format de pixel carré et redimensionnez au besoin votre vidéo.

Avec ou sans trame ? La vidéo analogique est composée de deux images (trame paire et trame impaire). Dans une diffusion pour la télévision, il faut toujours synchroniser ces deux images afin de ne pas donner l'impression d'un tremblement. L'image numérique n'est composée, elle, que d'une seule trame par image, mais intégrale. Si vos images proviennent de sources analogiques (caméscopes, VHS et/ou autre support Hi8), vous devez convertir ces images en images dites progressives, c'est-à-dire sans trame. Si, à l'inverse, vos images proviennent de sources numériques, c'est inutile : elles sont déjà progressives.

Notez que les logiciels de montage et de trucage, ainsi que l'encodeur Adobe livré avec Flash, permettent de convertir d'un format vers l'autre.

Une capture progressive vous permettra aussi d'extraire plus facilement une image nette, non tramée, pour la traiter dans Photoshop, en vue éventuellement de l'intégrer dans une composition Flash ou vers un document imprimé.

Quelle cadence pour le Web ? La cadence (ou fréquence) d'images est le nombre d'images fixes affichées en une seconde de film (elle est à 25 images par seconde pour une vidéo standard en PAL). Si la cadence pour la télévision ou le cinéma est importante, elle peut être réduite pour le Web. Mais ceci n'a pas réellement d'impact sur le poids des fichiers, contrairement aux idées reçues. En effet, les algorithmes traitent essentiellement des images-clés par nature insensibles à une variation légère de la cadence des images, sauf pour des mouvements réellement très marqués. Dans ce cas, le nombre d'images-clés devient plus important et leur itération plus serrée sur l'échelle du temps. Une modification importante de la cadence peut sans doute, dans ce cas, alléger un peu le poids d'une vidéo.

Mais indépendamment du poids des vidéos, la cadence agit surtout sur les performances d'affichage de la carte graphique. Plus il y a de variations à gérer dans un même temps donné, plus le processeur vidéo est sollicité.

Si la vidéo est importée physiquement dans le scénario de Flash, sa cadence doit être identique à celle de la vidéo. En revanche, si la vidéo est stockée en dehors de l'animation Flash, et que celle-ci est affichée dynamiquement dans Flash à l'aide du composant FLVPlayback

ou avec les commandes de la classe `NetStream`, sa cadence peut être différenciée. Lorsqu'une vidéo est importée dans Flash, si la cadence est différente de celle du scénario, le son, les animations et les actions risquent en effet d'être désynchronisés par rapport à l'image.

Combien de vidéos en simultanée ? Il est possible de multiplier les couches de fichiers vidéo au sein d'une même animation Flash, mais préférez les rassembler autant que possible dans un seul fichier vidéo. Vous gagnerez en fluidité.

Quelles dimensions pour une bonne image ? À débit équivalent, plus la vidéo occupe de la surface à l'écran, plus la compression devra être forte. Pour compenser le poids induit par les dimensions de la vidéo et stabiliser le débit, préférez réduire les dimensions des vidéos, vous obtiendrez un meilleur rendu. Dans Flash, les vidéos ne requièrent pas d'être importées dans un format standard de ratio 4/3 ou 16/9 par exemple. Les vidéos peuvent être recadrées avant l'encodage, de sorte que seule la surface utile à coder soit traitée sous la forme d'un signal vidéo utile. Par exemple, pour afficher un personnage qui marche en direction de l'utilisateur, une vidéo de forme rectangulaire verticale peut largement suffire. Le reste du décor sera traité en image fixe, réparti sur d'autres calques, dans Flash. Pour laisser voir le décor derrière le sujet filmé, il suffira de l'exporter avec une couche transparente (alpha) et de l'enregistrer au format FLV avec les options ON2 VP6 et canal alpha activées.

Images fixes ou en mouvement ? Plus la vidéo est riche d'un point de vue graphique (nous disons qu'elle possède du bruit), plus son poids sera conséquent, car c'est ce qui bouge qui est codé dans les algorithmes de compression. Ne rendez mobile que ce qui est utile ou alors, privilégiez les animations courtes.

Il est possible, dans certains logiciels de montage vidéo, de réduire le bruit sans trop altérer l'image. Étudiez cette option qui peut réduire considérablement la perte de qualité à l'encodage, ou, à qualité égale, réduire grandement le poids de la vidéo.

Capter en standard ou en HD ? Pour réaliser des montages vidéo simples, sans détourage ni incrustation, une capture standard suffit (PAL 4/3 758 × 576). Mais si vous souhaitez réaliser des trucages avancés, tels que le détourage d'un sujet capturé sur fond vert, préférez une image haute définition (1 080 lignes), dont le nombre de points et la qualité de compression aideront à affiner le détourage, avant de réduire l'image ensuite à l'exportation vers Flash.

Quel que soit le format de capture, songez que la vidéo doit être contenue, au final, dans une fenêtre de navigateur de surface utile de 980 pixels par 550 pixels.

La lecture est-elle instantanée ? Il faut compter une mise en cache de quelques secondes avant que la vidéo ne puisse être jouée lorsqu'elle est diffusée sur le Web. Prévoyez cela dans la scénarisation de votre document Flash. Cette propriété peut toutefois être contrôlée par `ActionScript`

Intégrer ou non les vidéos dans le Flash ? La vidéo pour le Web est de préférence externalisée par rapport au document Flash qui s'y réfère. C'est le Flash qui fait référence au fichier vidéo *via* une instruction `ActionScript`. Cela permet d'alléger considérablement le poids du document Flash et en simplifie la maintenance. En outre, la vidéo peut être lue à

mesure de son chargement, ce qui n'est pas le cas d'un document qui intègre physiquement la vidéo.

Lorsque la vidéo est incluse physiquement dans le Flash, il faut en effet attendre le chargement complet du Flash avant de pouvoir lire le début de la vidéo. À quelques exceptions près (que nous abordons au Chapitre 8, section "Lire une vidéo en arrière"), la vidéo est toujours externalisée.

Une vidéo chargée dans le scénario implique également la recompilation du film à chaque publication, ce qui prend en outre beaucoup de temps.

Le format vidéo de Flash peut-il être lu sans Flash ? Le Flash gère les formats vidéo FLV (version 6 et ultérieur) et F4V (version 10 et ultérieur), voire certains formats QuickTime. Mais les vidéos Flash ne sauraient être lues indépendamment du document Flash qui les accompagne, car c'est lui qui contient le lecteur en mesure de gérer ce format. Quelques applications toutefois savent lire les formats vidéo Flash, comme le lecteur VLC, Real, Bridge et certaines versions de Quick Time. Le format F4V, que nous détaillons au Chapitre 7, est plus permissif cela dit que le FLV, du fait de son type de compression, basé sur le H-264.

Source compressée ou non compressée ? Préférez toujours exporter au format vidéo de Flash à partir d'une source non compressée. Les vidéos déjà compressées utilisent des algorithmes qui peuvent générer des parasites dans la vidéo de sortie. Une vidéo non compressée demeurera toujours de meilleure qualité au premier encodage.

Quelle configuration pour lire une vidéo sur le Web ? Pour lire une vidéo gérée *via* Flash, sur le Web, l'utilisateur requiert au moins l'ADSL ou une connexion Câble, un lecteur Flash de version 6 pour les vidéos FLV basiques, de version 8 pour les vidéos FLV composites avec alpha, et de version théoriquement supérieure à 10 pour la vidéo haute définition avec le format F4V. Dans tous les cas, une bonne carte vidéo est souhaitable, supérieure ou égale à 64 Mo. En dessous de 64 Mo, les flux vidéos paraîtront parfois saccadés pour des tailles même raisonnables. En dessous de 32 Mo, elles paraîtront saccadées quelle que soit leurs dimensions, même en réduisant la cadence à 12 ou 15 images par secondes.

Quelle différence entre un flux continu et un chargement progressif ? L'affichage d'une vidéo à chargement continu est traité à partir d'un serveur de streaming (littéralement, d'envoi de flux) qui fait varier la quantité et la qualité des données en fonction de la bande passante disponible, dynamiquement. Les paquets de données vidéos étant traités dynamiquement, ce système permet de modifier la nature du contenu selon la configuration et les interactions de l'utilisateur. Cette technique permet ainsi la diffusion d'émissions en direct et l'envoi d'un signal plus ou moins compressé selon la configuration matérielle du poste utilisateur. Le traitement d'un signal continu *via* Flash implique une solution serveur telle que Flash Media Server.

Une vidéo à chargement progressif est un fichier déjà enregistré au format FLV ou F4V. Le lecteur Flash la copie dans le cache du navigateur et lit la vidéo, à mesure de son chargement. Le traitement d'un flux en chargement progressif ne permet pas le direct, mais reste très simple à déployer, car aucun serveur spécifique n'est requis pour lire la vidéo. C'est cette dernière technique que nous utilisons dans cet ouvrage.

Quelle différence entre un fichier FLV et un fichier F4V ? Le FLV propose une compression forte au détriment de la qualité, mais il autorise une implémentation souple à l'intérieur d'une mise en forme graphique composée de multiples calques. Cela en favorise l'interactivité. Le FLV encodé en On2VP6 autorise par ailleurs la transparence. Étant disponible depuis Flash 6, il apporte également une grande compatibilité.

Le format F4V utilise une compression HD (codec H-264). Il est utilisé pour l'affichage de présentations vidéo linéaires et qualitatives (bandes-annonces, teasers, produits de luxe, animations 3D avec de la radiosité). Une vidéo F4V est théoriquement seulement compatible avec Flash 10 et ultérieur, mais nous démontrons qu'elle peut être lue depuis une version plus ancienne au chapitre suivant. Ce format est privilégié pour les présentations haut de gamme et pour une cible bien identifiée qui sera équipée pour accéder à ce type de contenu (qui dispose d'un lecteur Flash récent et d'une bonne carte vidéo).

Composition simple avec Apple Motion

Apple Motion est un logiciel de trucage vidéo et d'effets spéciaux inclus dans la suite Final Cut Studio d'Apple. Cet outil permet de réaliser, entre autres, des habillages graphiques et animés à partir de formes vectorielles, d'images bitmaps, de particules générées dynamiquement et de séquences vidéo. Il est utilisé en télévision pour les habillages d'émissions, pour l'interfaçage de DVD et depuis peu pour le Web où il apporte une vraie touche graphique grâce notamment à son puissant moteur de génération de particules et d'accélération.

Dans cette section, nous allons voir comment exporter, depuis Motion, une animation de particules déjà prédéfinie, dans un format compatible avec l'encodage pour Flash (voir Figure 6.1). Dans cet exemple, l'animation de particules illustre l'éclat d'un feu d'artifice. Nous l'encoderons plus tard, avec sa transparence, pour Flash, et la placerons au-dessus d'une interface.

Figure 6.1

Aperçu de la vidéo Apple Motion après publication.





Exemples > videoMotion > particules.motr

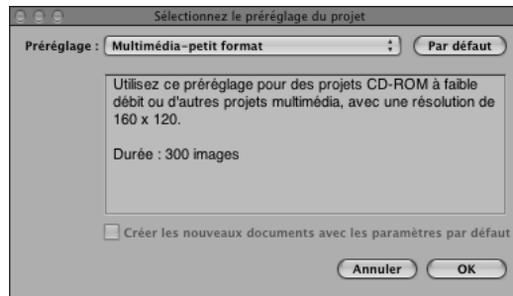
L'animation de particules que nous présentons ici est intégrée au logiciel Apple Motion. Pour les lecteurs qui ne disposeraient pas de l'application, une version exportée au format Quick Time est accessible dans le répertoire "videoMotion" des exercices du livre et se nomme "particules.mov". Vous pouvez directement passer à la section sur l'encodage si vous souhaitez intégrer des contenus vidéos FLV dans Flash.

Pour créer une animation vidéo de particules avec de la transparence, pour Flash et avec Motion, procédez comme suit :

1. Lancez l'application Apple Motion.
2. À l'ouverture, une boîte de dialogue de création de nouveau projet apparaît (voir Figure 6.2). Dans le menu Préréglages, sélectionnez l'option Personnaliser...

Figure 6.2

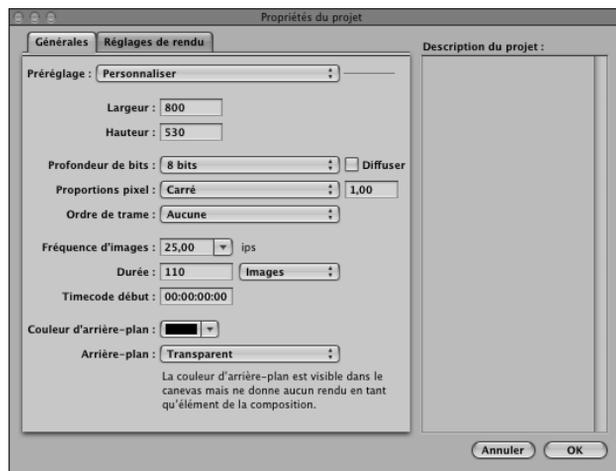
Fenêtre de nouveau projet dans Motion.



3. La fenêtre de Propriétés du projet apparaît (voir Figure 6.3).

Figure 6.3

Fenêtre de Propriétés du projet dans Motion.



4. Configurez un projet de résolution de 800 pixels de haut sur 530 pixels de large (ce qui correspond à notre surface de travail dans le document Flash), codé sur 8 bits (à savoir

la profondeur disponible sur les écrans informatiques tout public), de proportions de pixels Carré (soit un ratio de 1,00), sans trame. L'exemple que nous utilisons dure environ 110 images (durée de l'animation prédéfinie) et nous le cadencions à 25 images par secondes (25 ips, cadence standard pour la vidéo PAL en Europe). Nous sélectionnons enfin un arrière-plan de type Transparent, pour permettre plus tard la superposition de la vidéo avec d'autres éléments graphiques, au sein de l'application Flash.

Dans le nouveau projet, à gauche, figure la fenêtre Navigateur (voir Figure 6.4). Au centre, nous pouvons voir l'espace de travail, la scène. En haut, différents menus donnent accès à des effets qui peuvent être appliqués au contenu.

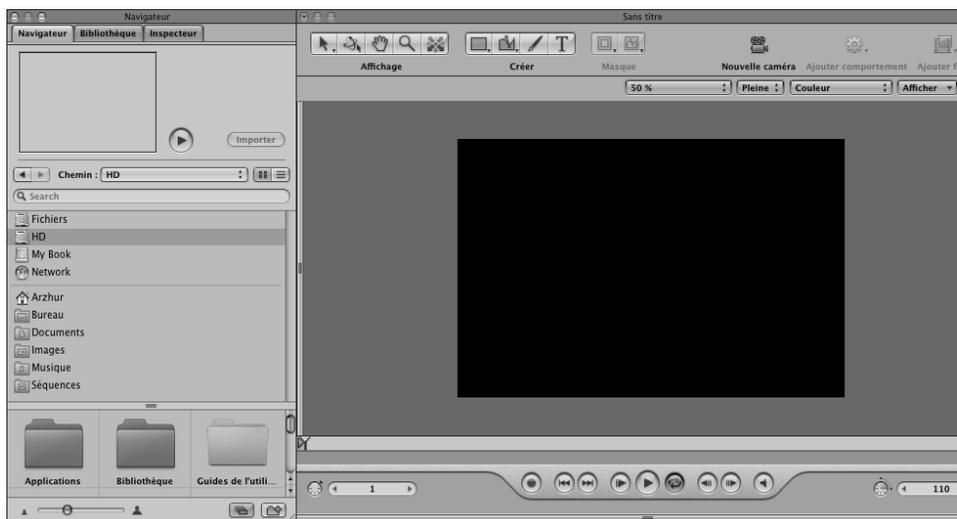


Figure 6.4

Fenêtre Navigateur.

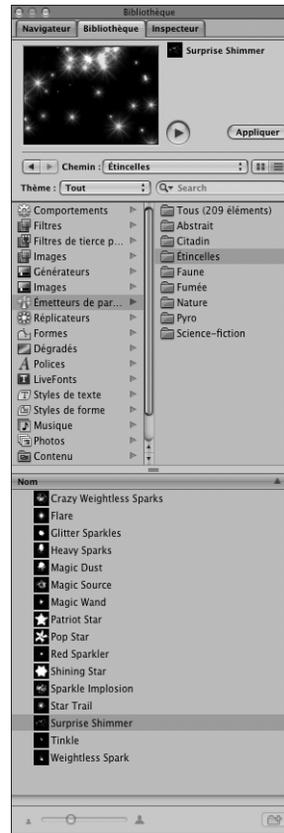
Dans la fenêtre Navigateur, vous pouvez identifier trois onglets : Navigateur, Bibliothèque et Inspecteur. Le Navigateur donne accès aux fichiers et dossiers de votre système. La Bibliothèque met à disposition un certain nombre d'objets prédéfinis et prêts à l'emploi. L'Inspecteur, lui, affiche toutes les options de contrôle disponibles pour l'objet activé dans la scène (comme l'Inspecteur de propriétés de Flash).

Cliquez sur l'onglet Bibliothèque. Puis, dans la partie inférieure, sélectionnez le répertoire Émetteur de particules. Dans la colonne de droite, sélectionnez maintenant le dossier Étincelles. Puis, dans la partie inférieure de la fenêtre, cliquez sur l'animation prédéfinie nommée "Surprise Shimmer" (voir Figure 6.5).

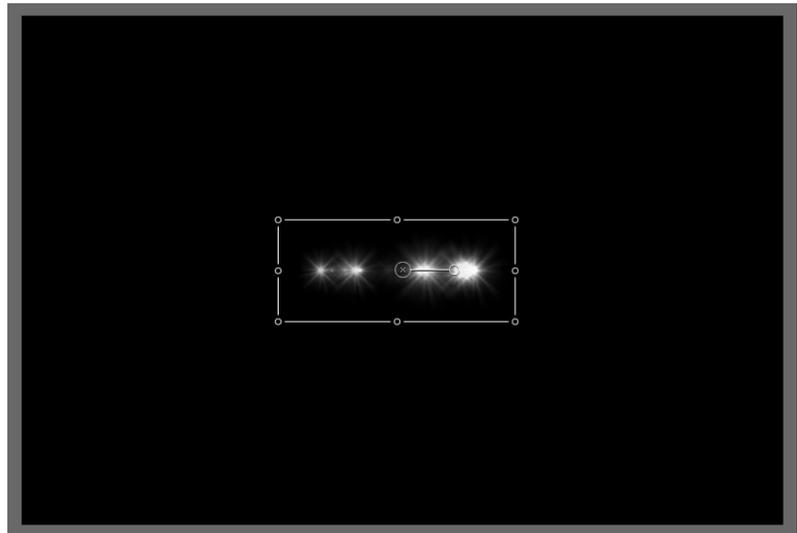
Un aperçu de l'animation est disponible au sommet de la fenêtre Navigateur. En bas de l'écran, vérifiez que la tête de lecture du scénario est bien située au début de l'animation (voir Figure 6.7).

Figure 6.5

Sélection d'une animation prédéfinie.

**Figure 6.7**

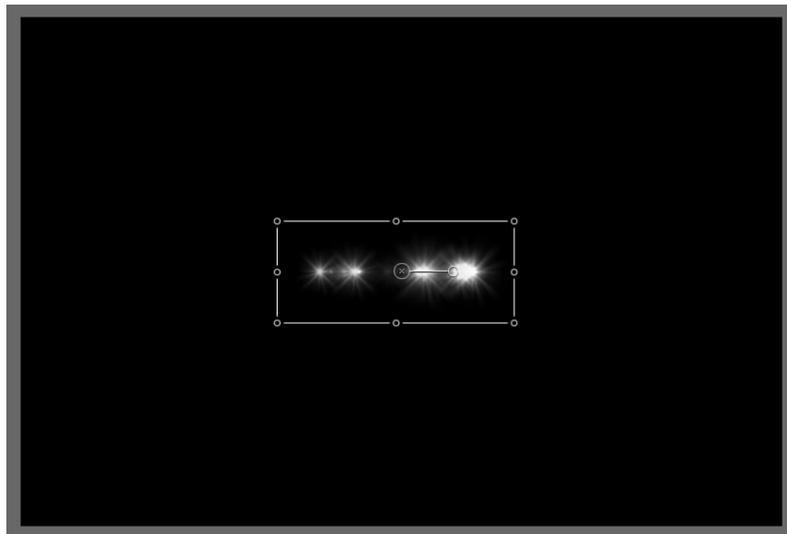
Fenêtre de scénario.



Glissez-déposez l'animation sélectionnée de la fenêtre Navigateur vers le centre de la scène (voir Figure 6.7). Puis, appuyez sur la barre d'espace pour tester l'animation.

Figure 6.7

Animation déposée.



Un scintillement d'étoiles s'anime dans la scène courante. L'animation fonctionne. Vous pouvez enregistrer le document de travail et l'exporter vers un format compatible pour l'encodage Flash.

Faites Fichier > Enregistrer. Puis, confirmez l'enregistrement du document Motion au format natif (.mtn) dans le dossier de votre choix. Par exemple, dans le répertoire Sources ou dans le dossier *videoMotion* des exercices du livre. Ce fichier natif ne sera pas mis en ligne. Il servira uniquement à modifier éventuellement votre création avant de l'exporter à nouveau dans un format aplati et compatible.

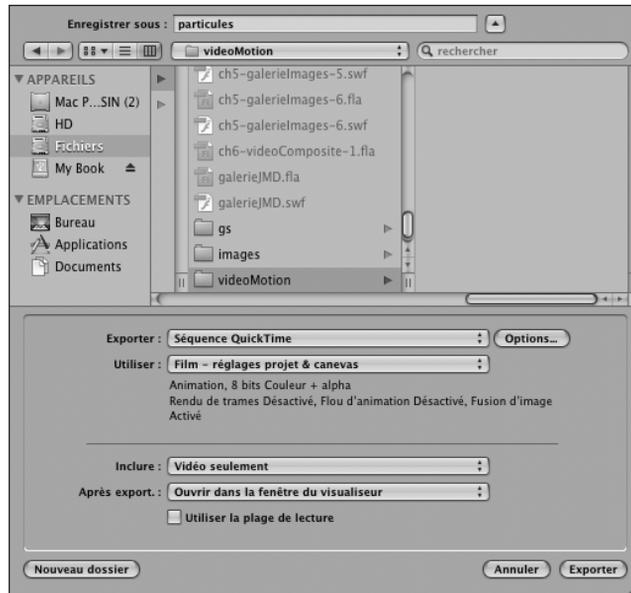
Vous pouvez maintenant exporter la vidéo pour le Web.

Dans Motion, il n'y a pas d'exportation directe au format Flash FLV ou F4V. Pour encoder la vidéo au format Flash, nous utilisons l'application Adobe Media Encoder. Mais, pour ce faire, nous devons d'abord exporter la vidéo dans un format standard, aplati et qui préserve la transparence. Nous choisissons de l'exporter au format usuel Animation de Quick Time :

1. Faites Fichier > Exporter. Puis, dans la boîte de dialogue, nommez la vidéo Particules. Plus bas, dans le menu Exporter, choisissez Quick Time. À droite, cliquez sur le bouton Options (voir Figure 6.8).
2. Dans les options d'exportation, choisissez une compression de type Animation en préservant au maximum la qualité de l'animation (voir Figure 6.9). Vous remarquez, dans le détail des paramètres de compression à droite, que la couche alpha est automatiquement intégrée dans ce format. Validez toutes les fenêtres. Puis, quittez l'application.

Figure 6.8

Fenêtre Exporter.

**Figure 6.9**

Options de compression.



La vidéo est maintenant exportée au format Quick Time et peut être encodée pour Flash avec Adobe Media Encoder. Reportez-vous, plus loin dans ce chapitre, à la section consacrée à l'enregistrement avec l'encodeur Adobe pour publier la vidéo au format Flash FLV.

Un autre exemple est disponible dans le dossier des exemples du livre et se nomme "galaxie.mov". Le fichier Motion associé se nomme "galaxie.mtn".

Encoder en FLV avec Apple Compressor

Compressor est livré en standard dans la suite Final Cut Studio de Apple. Pour les utilisateurs de Compressor, vous pouvez encoder au format Flash vidéo directement depuis le logiciel Compressor. Vous devez, depuis Compressor, exporter la vidéo au format Quick Time, et dans les options d'encodage du format Quick Time, sélectionner le format FLV.

Vidéo

À retenir

- Apple Motion permet d'exporter au format Flash vidéo FLV avec de la transparence, à condition de le publier d'abord au format Quick Time Animation, puis de l'encoder ensuite avec l'encodeur Adobe au format FLV.
- Apple Motion complète Adobe After Effects en cela qu'il intègre un puissant moteur d'interpolation des images et un bon générateur de particules.
- Les vidéos Apple peuvent également être exportées *via* Compressor pour Flash, avec le même paramètre d'exportation Quick Time pour l'animation, mais l'option FLV est ici disponible.

Composition simple avec Adobe After Effects

Comme nous l'avons vu avec Motion, nous allons voir comment exporter une animation simple réalisée ici avec After Effects, pour Flash.

After Effects est un logiciel de composition et d'effets spéciaux, disponible dans la suite vidéo Adobe. Ce logiciel, similaire à Motion, permet de réaliser des animations spatiales en 3D de manière plus avancée que Motion puisqu'il peut intégrer des objets modélisés en 3D, mais il ne dispose pas d'un moteur de particules aussi gratifiant. Les deux applications se complètent donc assez bien dans la production d'effets visuels. Il est à ce titre possible d'importer une vidéo réalisée par Motion dans After Effects et inversement, pour l'enrichir des fonctionnalités de l'autre application.

Dans cette section, nous allons voir comment créer une animation et la publier directement au format FLV ou F4V pour Flash, depuis After Effects.



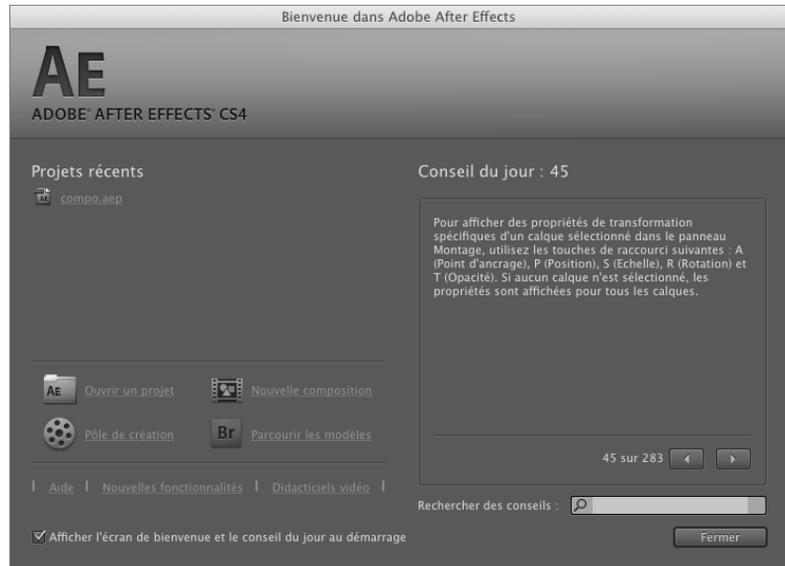
Exemples > videoAfterEffects > flocons.aep

Pour les lecteurs qui ne disposeraient pas de l'application After Effects, une version exportée de la vidéo est disponible dans le répertoire videoAfterEffects et se nomme flocons.mov. L'animation que nous allons étudier représente une pluie de flocons de neige déjà mise en forme. Mais avant d'ouvrir ce document, nous allons voir comment configurer un projet After Effects pour Flash :

1. Pour créer un document After Effects pour le Web, lancez l'application After Effects.
2. Au démarrage de l'application, une fenêtre d'accueil apparaît (voir Figure 6.10).

Figure 6.10

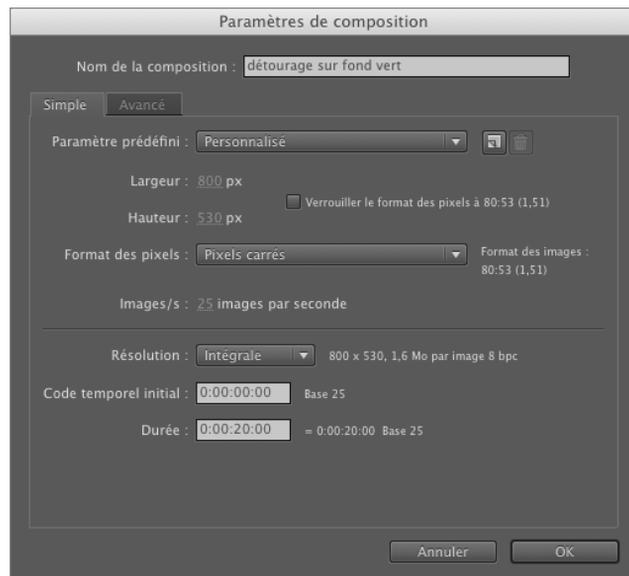
Fenêtre d'accueil d'After Effects.



3. Cliquez sur Nouvelle composition pour créer une nouvelle séquence de vidéo composite.
4. Une boîte de dialogue apparaît (voir Figure 6.11).

Figure 6.11

Paramètres de composition.



5. Dans la boîte de dialogue des paramètres de la nouvelle composition, spécifiez une configuration personnalisée de largeur 800 pixels et de hauteur 530 pixels (dimensions disponibles dans notre document Flash). Puis, choisissez un format de pixels carrés et une cadence à 25 ips, comme vu précédemment avec Motion. Spécifiez enfin une durée

souhaitée en secondes. Par exemple, pour une séquence d'une durée de 20 secondes, inscrivez 0:00:20:00 dans le champ correspondant. Validez.

After Effects affiche une nouvelle composition (voir Figure 6.12) basée sur des pixels carrés. Vous pouvez réaliser vos effets, importer des médias, multiplier les calques et animer les propriétés.

Figure 6.12

Nouvelle composition.



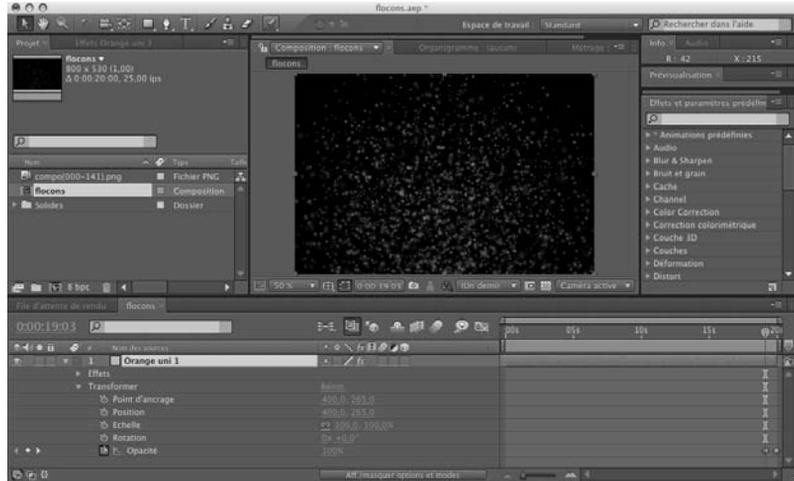
Dans le dossier des exemples du livre, une animation de flocons est déjà réalisée. Nous allons l'ouvrir pour l'exporter directement au format vidéo de Flash. Ouvrez le projet After Effects intitulé "flocons.aep", enregistré au format natif dans le dossier "videoAfterEffects".

Dans ce document (voir Figure 6.13), un solide est placé sur la scène et utilise un effet de particules (Effets > Simulation > CC Particles Systems II). L'effet neutralise le solide, mais l'utilise pour exister en tant qu'objet dans le scénario. Les options des effets sont accessibles dans le scénario et dans la fenêtre Effets (Fenêtre > Effets) et l'animation de fondu en sortie (fondu au noir) est matérialisée dans le scénario avec des images-clés appliquées à la propriété Opacité.

Pour exporter la composition directement au format vidéo de Flash, activez d'abord la séquence à exporter en cliquant sur la scène (où l'on aperçoit la vidéo). Faites Fichier > Exporter > Flash Vidéo FLV... Un message vous avertit que vous disposerez de plus d'options de contrôle en exportant depuis la fenêtre de rendu de After Effects. L'export *via* le menu Fichier utilise en effet une ancienne version de l'encodeur Flash vidéo, mais tout à fait conforme aux besoins d'une vidéo composite de base, avec alpha, et compatible Flash 8. Nous reviendrons sur les autres options dans la section "Échantillonner la vidéo pour Flash".

Figure 6.13

Composition
flocons.aep.



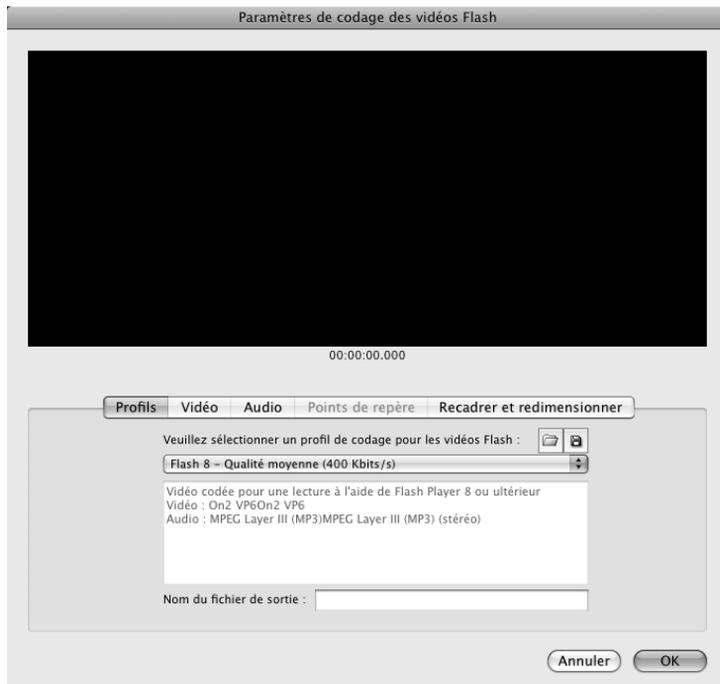
Validez pour confirmer l'enregistrement.

Dans la boîte de dialogue, vous pouvez accéder à différentes options d'échantillonnage (voir Figure 6.14), mais limitées.

Dans la liste des compressions prédéfinies située dans la partie inférieure de la fenêtre, vous accédez à différents formats d'encodage plus ou moins compatibles avec d'anciennes versions du lecteur Flash. Seuls les formats compatibles Flash 8 autorisent l'encodage de la couche alpha qui préserve la transparence de la vidéo dans Flash.

Figure 6.14

Paramètres
d'encodage des
vidéos Flash.



Pour un encodage qui autorise la transparence, sélectionnez Flash 8, dans la liste déroulante. Puis, adaptez votre choix selon l'option qui correspond le mieux à la bande passante définie pour votre cible. Reportez-vous au Tableau 6.1 pour connaître les valeurs à sélectionner en fonction de votre cible.

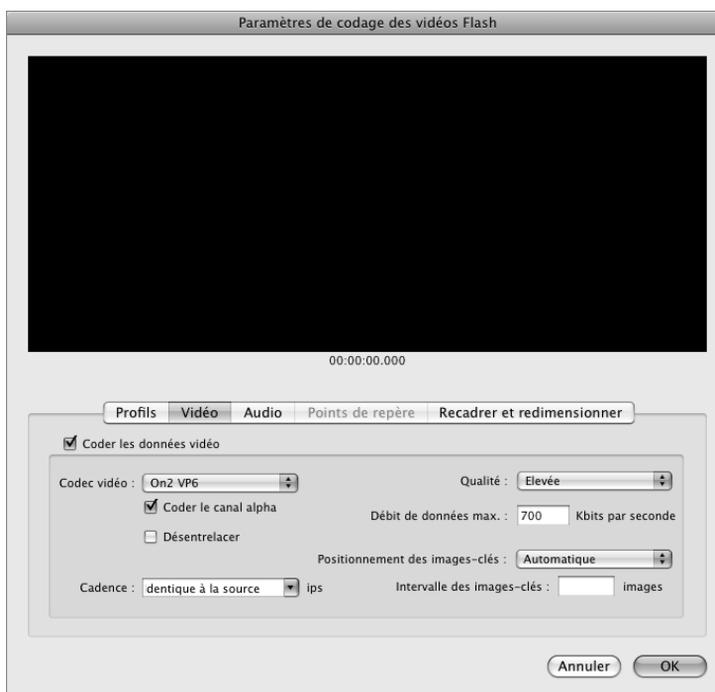
Tableau 6.1 : Compression et débit.

Débit	Compression
Utilisateurs Modem 56k	Pas de signal vidéo fluide.
Utilisateurs ADSL 1	400 kbps.
Utilisateurs ADSL 2 (majorité des utilisateurs)	700 kbps.
Utilisateurs Super ADSL, Câble, fibre optique	2 048 kbps à 4 096 kbps.
Utilisateurs locaux (non connectés, lecture en locale sur le poste utilisateur, CD, DVD)	> 4 096 kbps.

Dans la catégorie Vidéo (voir Figure 6.15), cochez l'option Coder le canal Alpha, puis validez.

Figure 6.15

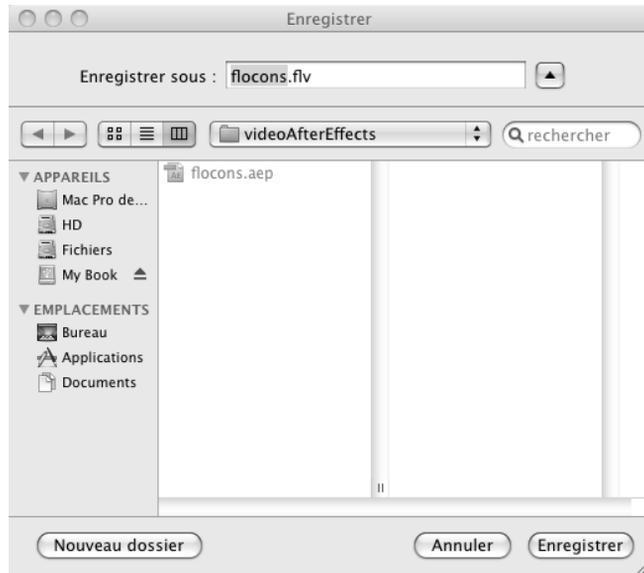
Coder le canal Alpha.



La fenêtre d'enregistrement apparaît. Nommez la vidéo flocons.flv, par exemple, puis validez (voir Figure 6.16).

Figure 6.16

Enregistrer la vidéo au format FLV.



La vidéo est enregistrée au format FLV. Vous pouvez l'intégrer dans Flash.

Pour le descriptif détaillé des options de compression du format FLV, reportez-vous à la section suivante. Pour savoir comment enregistrer pour Flash depuis d'autres logiciels vidéo (Final Cut Pro, Premiere Pro, mais aussi iMovie ou Window Movie Maker), reportez-vous au Chapitre 7.

Trucage avec captation sur fond vert

Si vous savez exporter une vidéo avec la couche alpha, vous savez aussi intégrer un sujet capturé sur fond vert. Pour réaliser une incrustation avec fond vert, relevez ce qui suit :

- Lors de la captation, veillez à obtenir un fond dont la couleur n'est pas spécifiquement verte, mais dont la teinte s'oppose en toutes circonstances à celle du sujet capturé.
- Utilisez un fond sans aspérité et régulier, qui réfléchisse bien la lumière en direction de la caméra et sans créer d'ombre ni laisser apparaître de plis.
- Rendez le fond homogène en l'éclairant sous plusieurs angles avec des lumières diffuses. Il ne doit pas y avoir de halo ou de zone moins exposée qu'une autre. Attention, une lumière trop forte génère un halo jaune (image brûlée).
- Éclairiez le sujet pour compenser le contre-jour que vous obtenez avec un fond exposé. Préférez un éclairage indirect du sujet, pour éviter l'apparition au sol ou sur le fond vert, d'ombres portées.
- Effectuez ensuite, si possible, un enregistrement en haute définition, avec de bonnes optiques, en image progressive et avec la compression la moins forte possible (donc, surtout pas en DV ni en HDV qui sont des formats très compressés), de sorte à restituer un piqué d'image le plus fin possible. Cela facilitera le détourage dans le logiciel de compositing.

- Pour détourer un fond vert, dans After Effects, vous pouvez utiliser les effets Keying > Keylight ou les effets du menu Effets > Correction colorimétrique, et les dégrossir par exemple avec un masque (pour supprimer d'abord les bords du cadre, les câblages qui entrent dans le champ, etc.). Dans Motion, dégrossissez de même avec un masque et utilisez Ajouter un filtre > Incrustation > Écran bleu ou vert pour le détourage.
- Réduisez éventuellement l'échelle de l'image, après le détourage bien sûr, juste avant de l'encoder au format vidéo de Flash.
- Encodez au format FLV avec l'option Coder le canal alpha.

Exporter le projet After Effects natif directement vers Flash

Si la composition After Effects doit utiliser des objets que vous souhaitez rendre interactifs dans Flash, vous pouvez exporter ces contenus directement vers Flash, sans rendre ni aplatir la vidéo. Faites Fichier > Exporter > Exporter au format Flash Professionnel XFL. Puis, depuis Flash, ouvrez le document XFL et enregistrez la composition au format FLA. Attention, tous les effets ne sont pas toujours bien interprétés. De plus, si des calculs doivent être appliqués à des animations de vidéos, un rendu sera malgré tout nécessaire avant de pouvoir importer le projet dans Flash. Le document Flash obtenu intègre physiquement la vidéo, ce qui en limite naturellement l'exploitation puisque nous savons qu'elle est mieux gérée en externe. Pour de plus amples informations néanmoins sur ces passerelles, consultez le livre *D'After Effects à Flash / De Flash à After Effects*, de Richard Harrington et Marcus Geduld, publié aux éditions Pearson, particulièrement adapté aux utilisateurs de Flash qui souhaitent par ailleurs apprivoiser After Effects.

À retenir

- Adobe After Effects offre une bonne transversalité avec Flash et permet d'exporter facilement au format FLV avec de la transparence.
- Il est possible d'exporter une composition After Effects directement vers Flash sans calcul de rendu, si les objets animés l'autorisent.

Échantillonner la vidéo pour Flash

Dans cette section, nous abordons l'encodage de fichiers vidéo, à partir d'une source déjà enregistrée (Quick Time, AVI, DV, etc.) ou depuis une application standard de création de contenu vidéo. Parmi celles-ci, nous abordons les solutions professionnelles des suites Adobe et Apple avec After Effects, Premiere Pro et Final Cut Pro, ainsi que les solutions embarquées sur les systèmes Mac et PC avec iMovie et Window Movie Maker.

À l'issue de ce chapitre, vous serez en mesure de convertir tout type de montage vidéo en format de fichier vidéo prêt à être intégré à Flash.