

RENCONTRES #EDMUS
TOULOUSE 14/03/2015

CRÉER UNE APPLICATION MUSICALE AVEC FLASH



@vieroux44



PRÉAMBULE

Tu viens d'installer Flash sur ton ordinateur, félicitations !

Ce merveilleux logiciel sert à créer des animations, des applications, et toutes sortes de contenus à destination du web... le tout encapsulé dans un unique fichier SWF. Que demander de plus ?

Astuce pour la suite des événements : le raccourci Ctrl + Entrée (Cmd + Entrée sur mac) sert à tester l'application au fur et à mesure de sa conception.

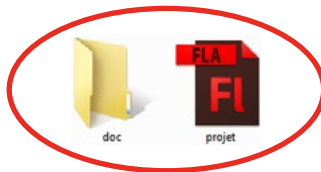
Le dossier qui t'est fourni contient la librairie *Greensock*. Celle-ci permet d'exploiter des dizaines d'animations prédéfinies avec une grande simplicité.

Nous nous servons pour cette initiation du dossier « *doc* » ainsi que du fichier « *Basics.swf* ».

D'autres fichiers sont également fournis (images et sons) : ils nous serviront de base pour créer un premier projet.

MISE EN PLACE

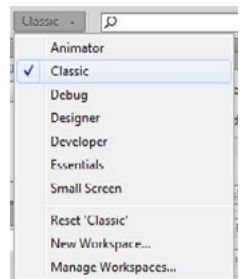
1. Crée un nouveau document de type « *Action Script 3.0* ».
2. À droite, indique la résolution du projet ainsi que le nombre d'images par secondes. Par exemple 800x600 à 24 FPS.
3. Enregistre aussitôt le fichier dans un dossier, et copie le répertoire « *doc* » au même emplacement.



L'INTERFACE

L'apparence du logiciel est entièrement personnalisable. Nous utiliserons aujourd'hui le mode « *Classic* ».

De nouvelles icônes peuvent être rajoutées dans les barres d'outils en les sélectionnant dans le menu « *Window* ».



IMPORT

Une fois que les bases du projets sont établies, il faut maintenant y intégrer nos différents ingrédients. Pour cela, deux options s'offrent à nous : glisser directement les fichiers sur l'espace de travail ou importer ceux-ci en passant par « *File / Import* » (Ctrl/Cmd + R).

Les fichiers arrivent alors dans la bibliothèque, où il devient possible de les organiser, trier, renommer, supprimer, etc.

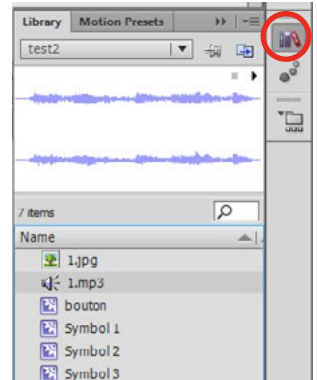


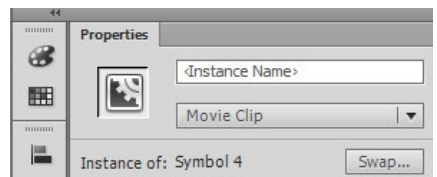
IMAGE VS. SYMBOL

Flash fonctionne avec plusieurs types d'images, à partir desquels les possibilités ne seront pas les mêmes. Il convient donc de savoir quels choix réaliser pour permettre à son application de fonctionner correctement.

Une image « ordinaire » ne sera pas prise en compte dans la programmation de l'application. Celles-ci seront réservées aux décors, ou à tous les autres éléments n'entrant pas en interaction avec l'utilisateur ou d'autres objets de la scène.

Afin de pouvoir programmer une image, il faut convertir celle-ci en « symbole ». Pour réaliser cette opération, fais glisser l'image dans la scène depuis la bibliothèque, clique-droit dessus et sélectionne « *Convert to Symbol* ». Dans la fenêtre qui apparaît, nomme cet objet afin de facilement le retrouver dans la bibliothèque, sélectionne le type « *Movie Clip* » et valide.

La dernière étape est la plus importante et permet de retrouver l'objet une fois dans l'interface de programmation : recopie ce même nom dans le panneau des propriétés.



INTÉGRATION DES SONS

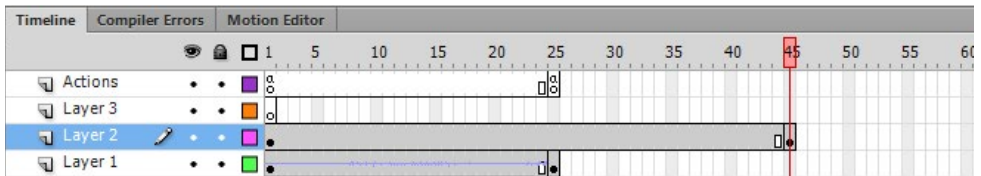
Même type d'opération pour les sons : ceux-ci doivent être « référencés ». Directement depuis la bibliothèque, clique-droit sur un son, sélectionne ses propriétés puis dans l'onglet « *ActionScript* », coche la 1^{ère} case et indique dans le champ « *Class* » le nom de l'objet. Il ne reste plus qu'à valider deux fois !

LA TIMELINE

Fonction fondamentale, la *timeline* va permettre de gérer le déroulement temporel de l'application. Selon la programmation établie, chaque colonne pourra correspondre à une scène entière (à durée parfois illimitée), ou bien à l'image individuelle d'une animation. Si le réglage initial est à 24 FPS, 24 colonnes totaliseront alors 1 seconde.

La *timeline* est composée de plusieurs calques (« *layers* »), qui vont permettre de gérer les différents plans de la composition. Les couches supérieures apparaîtront logiquement par dessous les couches inférieures.

Une ligne particulière est nommée « *Actions* » (si ce n'est pas déjà fait, à toi de la créer). Celle-ci est dédiée à la partie programmation.



Dans cet exemple, le deuxième calque est sélectionné (« *Layer 2* »)

Le curseur est situé à la toute fin (frame 45).

Les points noirs sur les calques 1 et 2 sont des « étapes », qui permettront par la suite de créer des transitions d'un état à un autre, d'une position à une autre, etc.

Finalement, les deux signes **a** sur la ligne « *Actions* » correspondent aux moments où une programmation manuelle intervient.

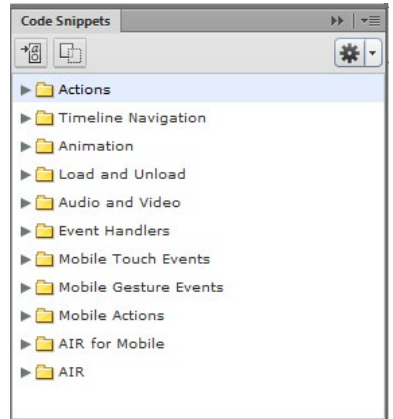
ENTRAÎNEMENT...

1. Sur un calque vide, clique-droit sur l'image n°1 et sélectionne « *Insert Keyframe* ». Une image-clé apparaît (point blanc).
2. Depuis la bibliothèque, fais glisser une illustration jusqu'à l'espace de travail. Le point blanc devient noir.
3. Recommence la première opération sur l'image n°25.
4. Déplace l'illustration à un autre endroit de la scène. En bougeant le curseur rouge de gauche à droite, tu remarques le déplacement instantané de l'image.
5. Maintiens la touche Shift enfoncée pour sélectionner les 25 images. Clique-droit sur la zone en surbrillance et sélectionne « *Create Classic Tween* ». Comme par magie, l'image se déplace maintenant progressivement !

SNIPPETS

Les « *Code Snippets* » sont des bribes de codes préprogrammées, qui permettent de gagner beaucoup de temps à la programmation, voire même de commencer à programmer sans même savoir comment s’y prendre. Ça tombe bien ! ;-)

Dans l’interface *Classic*, la fonction n’apparaît pas par défaut. Il faut donc aller la chercher dans le menu Window. On peut ensuite la faire glisser parmi les outils de la barre de droite.

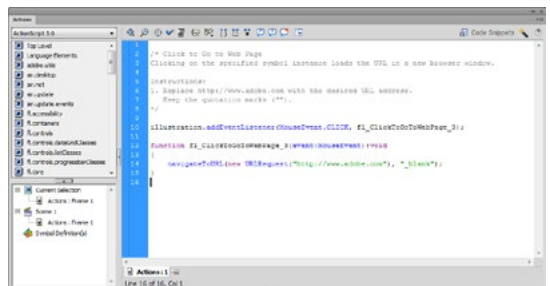


PREMIÈRES LIGNES DE CODE...

1. Insère une illustration (convertie comme symbole !) sur l’espace de travail, après avoir créé une image-clé (image n°1).
2. Si ce n’est pas le cas, vérifie que l’image est bien sélectionnée, puis ouvre le menu des *Snippets*.
3. Dans la partie *Actions*, nous allons double-cliquer sur la première option : « *Click to Go to Web Page* ».
4. Une fenêtre s’ouvre automatiquement, laissant apparaître le code-source correspondant.

ANALYSE ET PERSONNALISATION

1. La partie en gris correspond aux commentaires. Ces indications peuvent être supprimées, mais l’opération n’est pas forcément recommandée si l’on veut conserver un code facile à relire.
2. On y retrouve le nom de l’objet défini auparavant (ici, « illustration »).
3. On y voit de façon très claire l’adresse web qui s’ouvrira lors du clic sur l’objet.
4. À toi de transformer ce qui te semble transformable, et teste ton application (Ctrl/ Cmd + Entrée).



AUTRE EXEMPLE

Un peu plus complexe, l'exemple suivant nécessite de se griller 3 ou 4 neurones. Nous allons créer le scénario suivant : une image apparaît / l'utilisateur clique sur une illustration / la musique se déclenche / l'illustration initiale est remplacée par la suivante.

LES ÉTAPES

1. Insère une image-clé au n°1 et fais glisser une illustration (en tant que symbole) dans l'espace de travail.
2. Afin d'empêcher le programme d'aller directement à l'image n°2, utilise le *snippet* « *Stop at this Frame* ».
3. Crée un nouveau *snippet* sur la même image : « *Click to Go to Frame and Stop* ». Par défaut, la ligne de code suivante apparaît : `gotoAndStop(5)`. Ne souhaitant aller que vers l'image n°2, adapte le code à tes besoins.
4. Crée une image-clé au n°2 et remplace la première illustration par la seconde (toujours en temps que symbole). Teste l'application, il ne manque plus que le son !
5. Reviens sur la première image et applique le *snippet* « *Click to Play/Stop Sound* ».
6. Panique à bord, le code proposé est cette fois nettement moins lisible. Par défaut, Flash ira chercher un son de démonstration présent sur le web. Mais comme tu as déjà déclaré ton son d'après la procédure décrite à la page 3... (n'est-ce pas ?), il est possible de simplifier les choses.

On remplace l'austère :

```
var s:Sound = new Sound(new URLRequest(«http://www.helpexamples.com/flash/sound/song1.mp3»));
```

Par :

```
var s:Sound = new son();
```

(« son » étant dans cet exemple le nom déclaré au préalable)

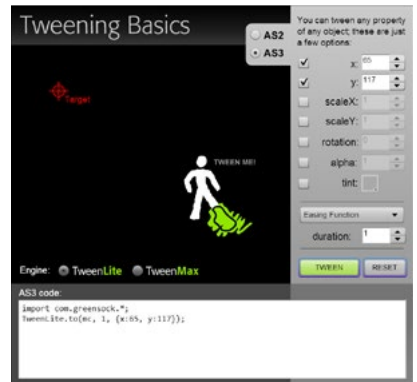
RÉÉDITER UN CODE

Tous les codes sont rééditables. Pour cela, clique sur l'image correspondante dans la ligne « Action » et appuie sur la touche F9.

DE QUELLE COULEUR SONT TES CHAUSSETTES ?

Nous en avons parlé au début... la fameuse librairie *Greensock*.

Comme évoqué à la page 4, il est possible d'animer des objets directement dans la *timeline*. Or, cette gestion est fastidieuse et ne permet pas facilement d'avoir une vue d'ensemble du projet. Bien souvent, il est judicieux d'animer un objet (symbole) au sein d'une même frame.



Pour rappel, le dossier « *doc* » et l'enregistrement préalable du projet au même emplacement sont indispensables au fonctionnement de la librairie.

EN VOITURE SIMONE !

Nous allons commencer par ouvrir le fichier « *Basics.swf* » à l'aide de n'importe quel navigateur web.

La procédure est la suivante : on déplace le bonhomme par rapport à la cible, on configure un mouvement dans le panneau de droite... et le code correspondant apparaît en bas comme par magie !

x et y	emplacement où l'objet va se rendre (depuis n'importe quel point de départ, que ce soit dans le champ ou hors-champ).
scale x et y	facteur d'agrandissement / rétrécissement.
rotation et alpha	rotation et degré de transparence.
tint	coloration de l'image.
Easing Function	liste des mouvements proposés.
duration	durée de l'animation
Tween	tester l'animation.

ÇA ROULE ! OUI, MAIS...

Par défaut, le code proposé transforme un objet dont le nom de symbole est « *mc* ». Ce qu'il faut bien entendu changer pour le nom de l'objet à programmer.

À TOI DE JOUER

Maintenant que tu as tous les éléments en main, à toi de créer le *serious game* de l'année.

Tu dois maintenant être capable de...

- Personnaliser le déroulement de l'application Flash, en redirigeant l'utilisateur vers un numéro d'image correspondant au bouton qu'il a cliqué.
- Déclencher ou arrêter une musique.
- Programmer toutes sortes de déplacements (à la main ou dans *Greensock*).
- Masquer ou afficher un objet en retrouvant les *snippets* appropriés dans le menu.
- Appeler une page web depuis l'application.

POUR ALLER PLUS LOIN

Essaie* de combiner/adapter/transformer le *snippet* « *Mouse Over Event* » avec un mouvement de *Greensock*, afin de faire faire un looping à un objet dès que la souris passe au-dessus de celui-ci.

Solution sur www.musikobreil.fr



*
N'essaie pas, fais-le !