Interface Builder
Project Builder
Niveau : intermédiaire

Vos premiers pas avec Cocoa

Yannick Cadin

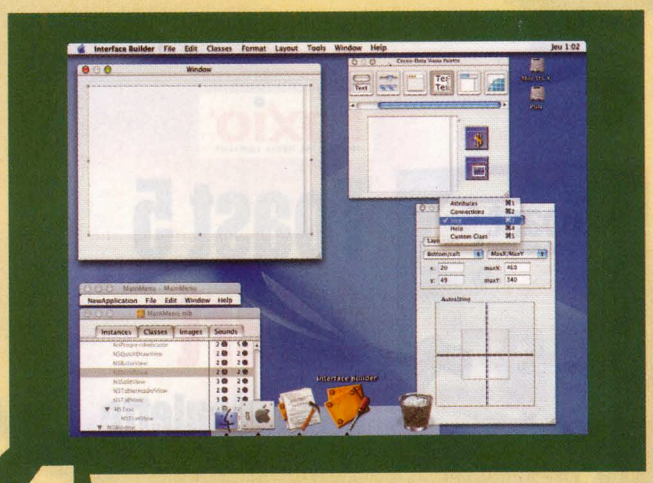
Il faut bien commencer un jour : voici notre premier Pas à pas consacré à la programmation sous Mac OS X ! Prenez les commandes... *Dans le système se trouve l'application Terminal, au sein du dossier Utilities. Lancez-la et tapez les lettres lsof, puis Entrée. Vous obtenez la liste des fichiers ouverts par l'ensemble des programmes. Cela vous permet, par exemple, d'identifier un fichier interdisant d'éjecter un volume. C'est cette commande lsof que nous vous proposons de transformer en une application de votre cru, plus conviviale. À l'aide de l'environnement de développement de Mac OS X (le troisième CD-ROM fourni avec le système), vous allez lui ajouter une interface graphique présentant une fenêtre d'affichage redimensionnable et un bouton pour mettre à jour le résultat. La programmation objet fait appel à un vocabulaire ésotérique, mais vous pouvez l'ignorer pour mener à bien ce Pas à pas. Cependant, si vous voulez éclaircir les termes "classe", "méthode" ou "instance", consultez le Coin de l'expert (page 122).*

Les ingrédients

- Project Builder et Interface Builder, tous deux livrés avec Mac OS X.0.

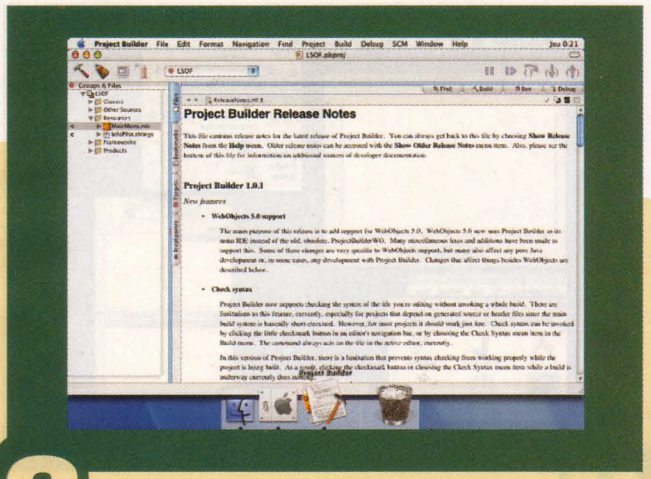
1 Définir le projet.

Lancez Project Builder, placé dans le dossier Applications du dossier Developer. En choisissant *New Project* du menu *File*, vous obtenez l'assistant de projet. Sélectionnez *Cocoa Application* pour lui signifier le procédé de programmation que vous allez utiliser, puis cliquez sur *Next*. Dans *Project Name*, saisissez le nom de votre future application, c'est-à-dire LSOF, et indiquez le dossier où il faut l'enregistrer : ~/LSOF/. Cliquez sur *Finish*. Si vous aviez voulu réaliser un programme qui gère des documents, comme Word ou Photoshop, vous auriez choisi *Cocoa Document-based Application* dans le premier écran. ■



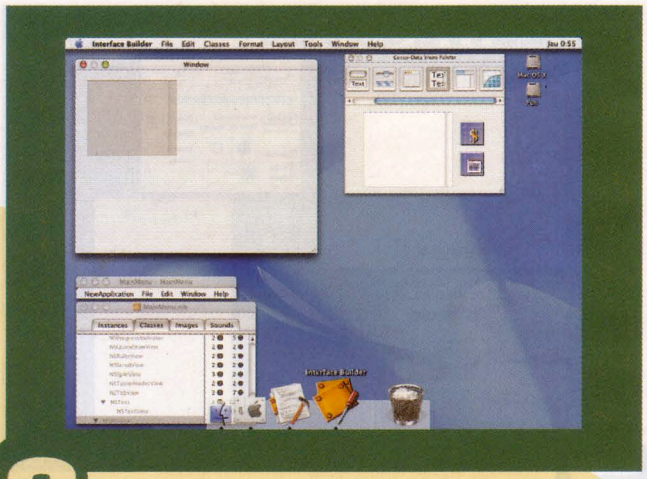
4 Utiliser l'Inspecteur.

Sélectionnez l'objet NSTextView et agrandissez-le afin qu'il occupe la majeure partie de la fenêtre *Window*. Laissez de la place libre pour un bouton en bas de cette fenêtre. Optez ensuite pour l'article *Show Info* du menu *Tools*. L'Inspecteur surgit. Il vous aidera à modifier les attributs et les comportements dynamiques de vos objets. Vous pouvez y déterminer la couleur du fond et celle du texte, ainsi que le type d'encadrement de l'objet concerné. Vous devez également définir de quelle façon les dimensions évolueront lorsque la fenêtre principale sera redimensionnée. Activez maintenant *Size*, dans le menu local en haut de l'Inspecteur. ■



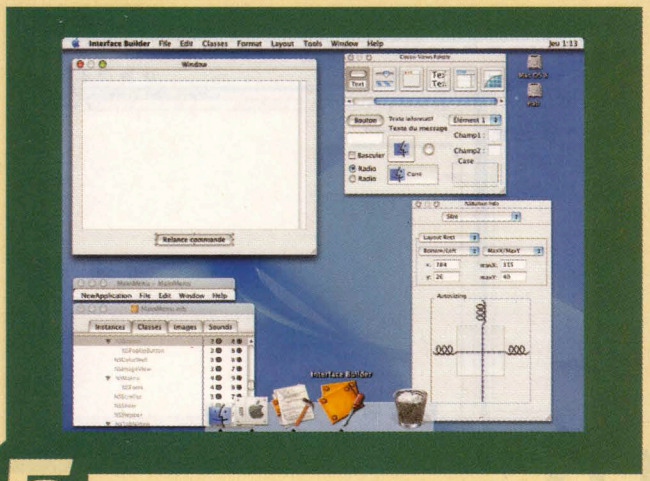
2 Ouvrir l'interface.

Vous voilà en présence de la fenêtre principale de Project Builder. La liste hiérarchique située dans la colonne de gauche présente les fichiers nécessaires au projet, son squelette en quelque sorte. Vous allez employer en particulier le fichier MainMenu.nib qui se trouve dans le dossier Resources. Ce fichier renferme un embryon d'interface que vous allez compléter avec les éléments requis par votre utilitaire. En l'ouvrant d'un double clic, vous provoquez le démarrage d'Interface Builder. Il charge le document désigné, ainsi que les objets par défaut qui constituent son environnement de travail. ■



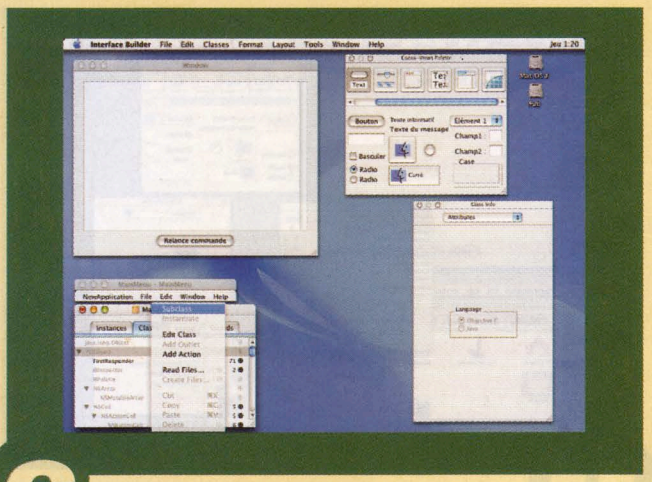
3 Compléter l'interface.

Le modèle de projet adopté au départ conditionne l'interface qui apparaît. Interface Builder affiche un menu sous forme de fenêtre, ainsi qu'une fenêtre vide. Tous les objets de votre projet sont représentés par des icônes dans l'onglet *Instances* de la fenêtre *MainMenu.nib*. La palette intitulée *Cocoa-Data Views* propose les objets que vous pouvez ajouter. Là, cliquez sur la quatrième icône en partant de la gauche, puis glissez-déposez le cadre blanc sur la fenêtre vierge *Window*. Il s'agit d'un objet NSTextView qui accueillera le résultat de la commande lsof. Le préfixe NS est un vestige de NEXTSTEP. ■



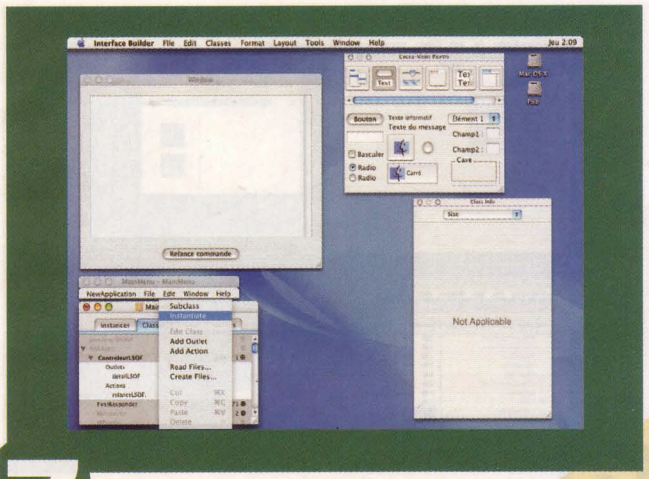
5 Insérer un bouton.

Les cadres intérieur et extérieur symbolisent respectivement l'objet sélectionné et celui où il est placé. Cliquez sur les lignes horizontale et verticale du cadre intérieur pour faire apparaître les ressorts, afin que NSTextView s'adapte à la taille de la fenêtre quand elle sera redimensionnée. Cliquez sur la deuxième icône en haut de la fenêtre *Cocoa-Data Views* et glissez-déposez l'objet Bouton en bas de la fenêtre *Window*. Double-cliquez dessus pour le rebaptiser "Relance commande". Dans l'Inspecteur, cliquez en haut, à droite et à gauche pour faire surgir les ressorts. Les dimensions du bouton ne changeront plus, et sa position par rapport aux bords droit, gauche et haut sera variable. ■



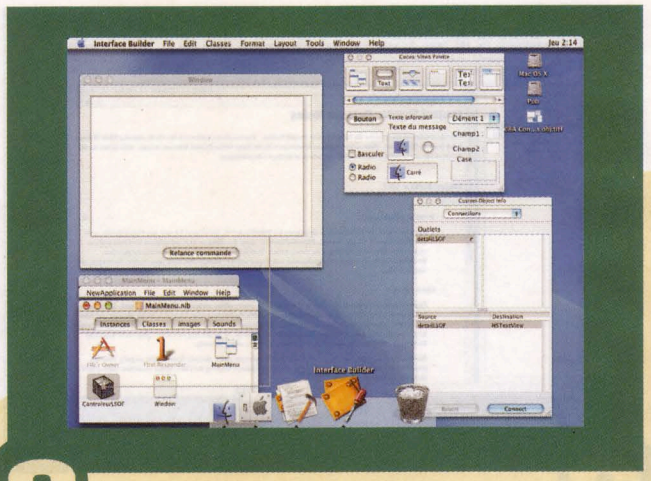
6 Créer une classe.

L'interface à proprement parler est terminée. Il faut maintenant relier ses éléments à l'objet qui contiendra les traitements. Vous allez d'abord générer un objet qui hébergera le cœur du programme. Dans la palette *MainMenu.nib*, sélectionnez l'onglet *Classes*, puis la ligne *NSObject*. Choisissez *Subclass* dans le menu *Classes*. Une sous-classe appelée MyObject fait son apparition ; renommez-la Controleur-LSOF. Vous allez devoir préciser sa variable d'instance (ou Outlet), par laquelle elle communiquera le résultat de ses opérations, ainsi que sa méthode d'instance (ou Action), par laquelle elle recevra l'ordre de procéder au calcul. ■



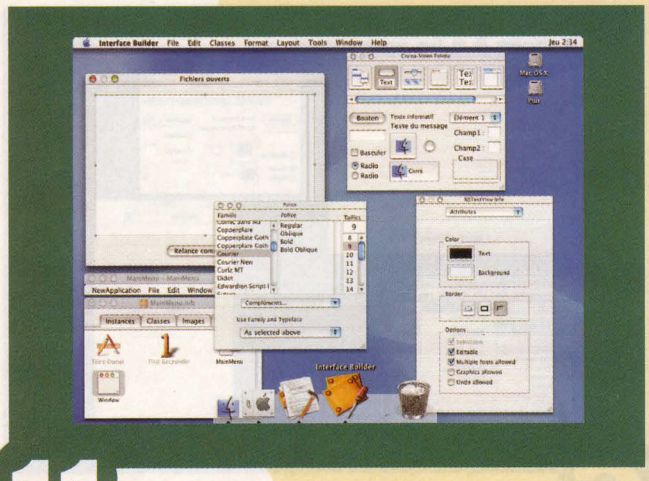
7 Déclarer la méthode.

Pour déclarer une variable d'instance, choisissez la classe ControleurLSOF, puis optez pour *Add Outlet* dans le menu *Classes*. Renommez-la "detailLSOF". Déclarez aussi la méthode relanceLSOF à l'aide de l'article *Add Action* du même menu. Les noms de classes commencent par des majuscules, ceux des variables et méthodes d'instance par des minuscules. À l'intérieur des noms, des majuscules marquent le début de chaque mot. Une fois les déclarations achevées, sélectionnez ControleurLSOF et choisissez *Instantiate* du menu *Classes* pour en générer une instance. C'est elle que vous lierez aux objets de l'interface. ■



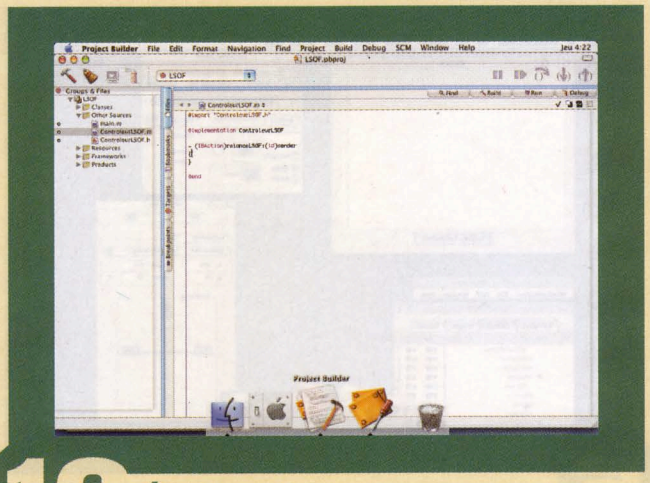
8 Associer des objets.

Cliquez sur *Instances* de la fenêtre *MainMenu.nib*. Vous y retrouvez votre instance de la classe ControleurLSOF. Gardez la touche Contrôle enfoncée pendant que vous tirez un lien depuis cette instance vers l'objet NSTextView de la fenêtre *Window*. Lorsque vous relâchez le bouton de la souris, l'Inspecteur montre la liste des variables d'instance du ControleurLSOF et vous demande de confirmer la connexion. Assurez-vous que detailLSOF est bien sélectionné, puis validez avec *Connect*. L'instance de la classe ControleurLSOF est maintenant associée à l'objet NSTextView par l'Outlet detailLSOF. ■



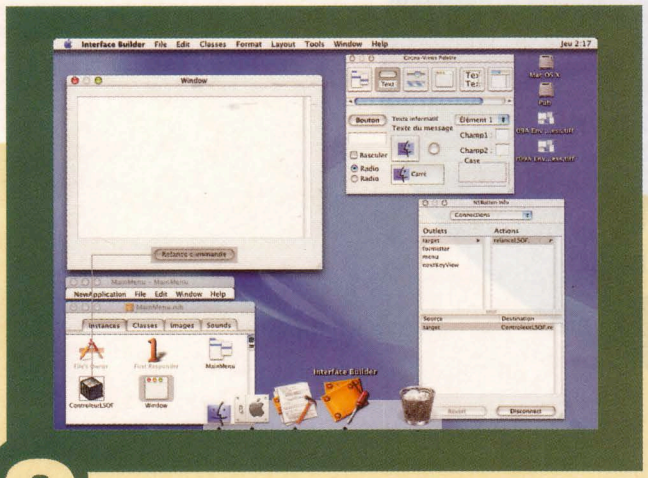
11 Finition de l'interface.

Pour renommer la fenêtre de votre application, sélectionnez l'icône Window dans l'onglet *Instances* de la fenêtre *MainMenu.nib*, puis saisissez "Fichiers ouverts" dans le champ *Title* de l'Inspecteur. Indiquez maintenant la police et le corps dans lesquels doit s'inscrire le contenu de votre objet NSTextView. Pour ce faire, choisissez cet objet dans votre fenêtre, puis faites appel à *Show Fonts* dans le sous-menu *Font* du menu *Format*. Adoptez la police Courier en corps 9. Contrôlez dans l'Inspecteur que la case *Editable* est cochée, afin que le programme ait l'autorisation d'écrire dans l'objet. Vous en avez terminé avec Interface Builder. ■



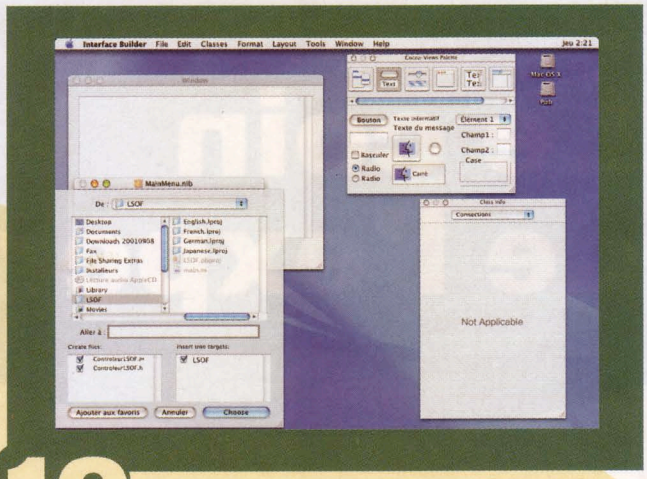
12 Écrire le code.

En retournant dans Project Builder, vous retrouverez dans la colonne de gauche les références aux fichiers de définition de votre classe ControleurLSOF. Le document ControleurLSOF.h possédant déjà toutes les déclarations nécessaires, il ne vous reste plus qu'à compléter le fichier d'implémentation ControleurLSOF.m. Sélectionnez-le, il surgit dans la fenêtre de droite. Seuls sont présents les points de départ de chaque méthode créée dans Interface Builder, en l'occurrence relanceLSOF. À vous de compléter ce code source par le cœur de votre programme. Sélectionnez tout et collez, à la place, l'intégralité du fichier code source, disponible sur notre CD-ROM. ■



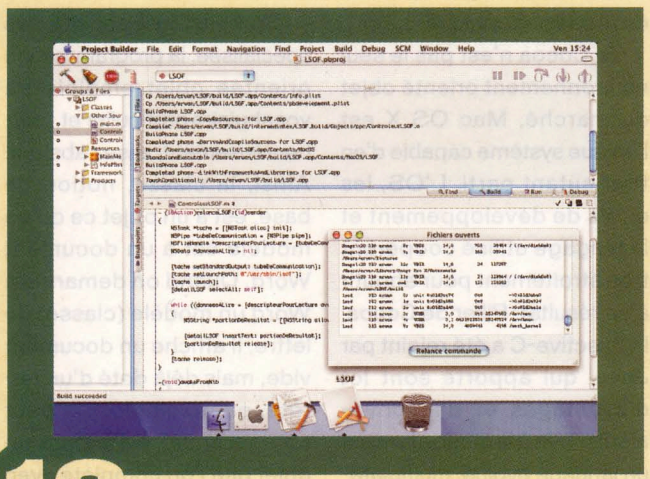
9 Envoyer un message.

On procède comme précédemment pour raccorder un objet actif de l'interface à l'une des Actions d'un autre objet. Il vous reste à établir la liaison entre le bouton et l'objet de type ControleurLSOF que vous avez conçu, en signalant qu'un clic sur le bouton doit entraîner un envoi de message à la méthode relanceLSOF. Pour ce faire, reliez les deux objets au moyen de la souris et de la touche Contrôle, puis sélectionnez tour à tour les éléments *target* et *relanceLSOF* dans l'Inspecteur. Cliquez ensuite sur le bouton *Connect*. Il ne s'agit pas ici du simple raccord de deux objets, puisque l'on précise aussi la méthode qui sera invoquée. ■



10 Sauvegarder des fichiers.

À partir des données que vous avez indiquées (nom, superclasse, noms des variables, méthode d'instance), vous allez demander à Interface Builder de générer deux fichiers avec les embryons de déclaration et d'implémentation de votre classe. Il suffit d'opter pour ControleurLSOF dans l'onglet *Classes*, puis *Create Files* du menu *Classes*. Le titre des fichiers reprend obligatoirement celui de la classe, avec les suffixes ".h" pour la partie déclaration et ".m" pour l'implémentation. Vérifiez que toutes les cases en bas du panneau d'enregistrement sont cochées. Choisissez le dossier LSOF, cliquez sur *Choose*. ■



13 Bâtir le projet.

Vous devez encore construire l'application, ce qui se limite, avec ce projet simple, à compiler le code source en Objective-C. Notez que l'interface décrite dans le fichier .nib n'est pas vraiment ajoutée au code exécutable, mais prend place, comme les autres ressources du programme (y compris le code compilé), au sein d'un dossier. C'est ce dernier qui est vu comme un fichier exécutable par Mac OS X. Le programme lit ce document .nib à chaque exécution. Choisissez *Build and Run* dans le menu *Build* pour lancer la fabrication de l'utilitaire. Le voilà à présent terminé. Il a pris place dans le sous-dossier Build du dossier LSOF. ■

Juste pour goûter

Le présent programme constitue une simple démonstration et demeure très perfectible. Notamment, la commande lsof génère de très nombreux résultats, dans lesquels il est difficile de se retrouver. Une commande de recherche ou de filtrage serait la bienvenue. Autre conséquence du généreux résultat de notre commande lsof : la mémoire tampon qui reçoit provisoirement le résultat d'une commande est insuffisante pour en contenir l'intégralité. Nous n'aurions pas rencontré ce problème avec une commande au retour d'information plus modeste. C'est la raison d'être de la boucle qui lit le buffer au fur et à mesure de son remplissage, jusqu'à ce qu'il soit vide. Malgré ses faiblesses, votre utilitaire est un "front end" efficace et adaptable. On en trouve un autre dans les utilitaires standard de Mac OS X : Network Utility qui donne accès, de la même manière, aux principales commandes de statistiques réseau d'UNIX, comme ping ou traceroute. Enfin, cela représente un bon point de départ pour réaliser des utilitaires plus ambitieux avec un certain nombre d'autres commandes UNIX. Le principe ne change pas et vous adapterez facilement le squelette employé ici. Mais il faut évidemment passer de longues heures devant la documentation HTML qui figure dans Developer/Documentation/.