

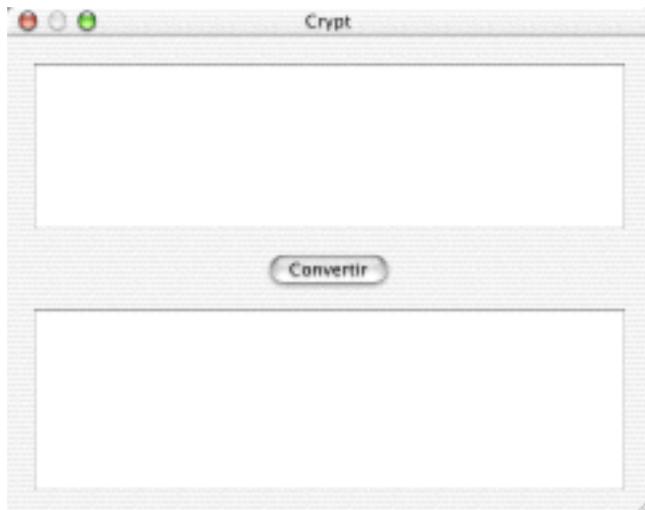
Construire l'application Crypt

Pré-requis : installer Mac OS X.1 avec les "developer tools" - voir installation sur le site.

1) Création du projet et de l'interface

Après avoir lancé ProjectBuilder (PB), sélectionner NewProject dans le menu Files, puis suivre la procédure décrite dans "Developping Cocoa Java Applications: A Tutorial" (voir l'accès sur le site), création du projet "Crypt" (Cocoa-Java Application), création d'une interface (double-clique sur MainMenu.nib dans le groupe Ressources du projet, provoque le lancement de InterfaceBuilder (IB)).

La fenêtre "Window" de notre application est vierge, nous allons y ajouter deux champs de text (NSTextField de la palette Cocoa-Views) avec entre les deux un bouton (NSButton de la palette Cocoa-Views).



Nous allons maintenant fixer les principaux attributs en taille des éléments graphiques au moyen de l'inspecteur (Show Info du menu Tools, écran Size). En sélectionnant la fenêtre, cliquer sur le bouton Current de Min Size (verrouillage de la taille minimale). En sélectionnant le champ texte du haut, cliquer sur les deux lignes internes en croix et sur la ligne externe du bas pour faire apparaître des ressorts. De même pour le champ texte du bas mais sur les deux lignes internes en croix et sur la ligne externe du haut pour faire

apparaître des ressorts. De même pour le bouton, uniquement les quatre lignes externes. Une ligne droite établit un lien fixe entre les deux éléments. Un ressort permet l'évolution indépendante des deux éléments. Notre fenêtre peut être maintenant redimensionnée dans tous les sens, sans mauvaise surprise, essayer en la testant (Test Interface du menu Files).

2) Création de la classe et connexion des objets

Nous allons maintenant créer une classe pilotant ces trois objets graphiques comme dans l'exemple, sélection de `java.lang.Object`, Subclass du menu Classes, nom de la classe "Crypt", création de deux objets (Outlets) "entree" et "sortie" (Add outlet du menu Classes) puis d'une action "convertir()" (Add action du menu Classes) dans l'inspecteur.



Revenons sur la classe Crypt et créons l'instance relais (Instanciate du menu Classes) qui va permettre de créer les connexions aux objets.

La connexion des outlets est effectuée en tirant la souris depuis l'instance vers l'objet graphique tout en maintenant la touche contrôle enfoncée. Cliquer sur le nom de l'outlet correspondant à l'objet graphique ("entree" pour la zone texte du haut, "sortie" pour la zone texte du bas) puis cliquer sur "connect".

La connexion des actions est effectuée en tirant la souris depuis l'objet graphique vers l'instance tout en maintenant la touche contrôle enfoncée.

Cliquer sur le nom de l'action correspondante à l'objet graphique ("convertir" pour le bouton) puis cliquer sur "connect".

Sauvegardez.

3) Création du code source

Revenons une nouvelle fois sur la classe Crypt et créons le squelette du code source (Create Files du menu Classes), cliquez sur "Choose".

Revenons à PB, Crypt.java apparaît, nous allons le positionner dans le groupe Classes et l'éditer en cliquant dessus. Les types des objets est à modifier avec le type réel de l'objet graphique (NSTextField).

Il ne reste plus qu'à remplir notre fonction convertir avec le code suivant:

```
/* Crypt */
import com.apple.cocoa.foundation.*;
import com.apple.cocoa.application.*;

public class Crypt {
    NSTextField entree;
    NSTextField sortie;

    int rot(int c) {
        if ((c >= 'A') && (c <= 'Z'))
            c = (((c - 'A') + 13) % 26) + 'A';
        if ((c >= 'a') && (c <= 'z'))
            c = (((c - 'a') + 13) % 26) + 'a';
        return c;
    }

    public void convertir(Object sender) {
        char [] texte = entree.stringValue().toCharArray();
        int ind;
        for (ind = 0; ind < texte.length; ind++)
            texte[ind] = (char) rot(texte[ind]);
        sortie.setStringValue(new String(texte));
    }
}
```

```
}  
  
}  
/* fin de Crypt */
```

3) Exécution

Il ne nous reste plus qu'à lancer l'exécution de notre application (Build and Run du menu Build).

Après avoir rempli le texte du haut, un clique sur le bouton Convertir fait apparaître le texte crypté dans le bas. Pour décrypter il suffit de copier le texte crypter dans le texte du haut et cliquer de la même façon sur Convertir. Cet algorithme est symétrique.

La sélection de Wraps (Layout dans l'inspecteur de IB) dans les attributs des zones de texte permet d'avoir le texte sur plusieurs lignes. Le retour à la ligne forcé s'effectue avec <opt><CR>.

Voici le point de départ vers des applications plus ambitieuses comme Astro proposée sur le site.

Pascal Pignard, novembre 2001.