

**Master Informatique 1ère année - Université Paris-Sud**  
**Ingénierie et Systèmes Interactifs**  
**Examen - 19 mai 2006 - 3h**

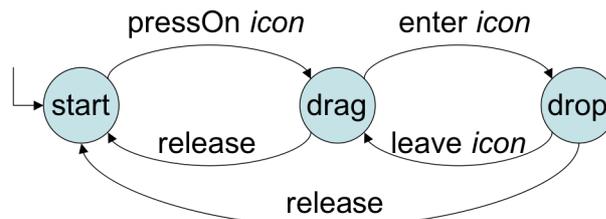
Seul document autorisé : feuille A4 recto-verso manuscrite.  
Lisez l'énoncé en entier. Soyez clairs, précis et concis.

**A. Questions de cours (6 points)**

1. Décrire les trois facettes d'un widget d'une boîte à outils d'interface et les illustrer avec deux widgets différents.
2. Décrire le modèle d'architecture MVC (Modèle-Vue-Contrôleur). Expliquer comment il permet de réaliser des vues multiples synchronisées.
3. Expliquer la différence entre feed-back et feed-forward et décrire en quoi ils peuvent améliorer l'utilisabilité d'une interface. Illustrer vos explications par deux exemples.
4. Donner quatre variantes de menus et expliquer leurs avantages et inconvénients respectifs.

**B. Modélisation de l'interaction (7 points)**

On considère l'interaction de type "drag-and-drop" dans une interface iconique, décrite par la machine à états ci-dessous :



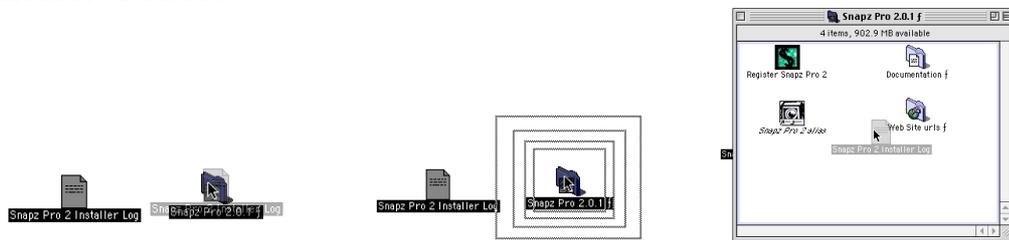
1. Compléter cette machine à états avec le code des actions et, si nécessaire, des états et transitions supplémentaires afin d'implémenter un drag-and-drop standard, à savoir :
  - feedback du déplacement ;
  - "disparition" de l'icône source dans l'icône destination si on relâche sur un icône, ou simple déplacement de l'icône source sinon.

**Note :** dans tout cet exercice on demande de définir les machines à états sous forme graphique et les actions avec du pseudo-code, pas du Java. Vous pouvez introduire des classes / fonctions / méthodes à condition de les définir avec assez de précision : soit une description informelle, soit du pseudo-code.

2. On enrichit l'interface ci-dessus avec des icônes de type "dossier" que l'on peut ouvrir pour présenter leur contenu dans une fenêtre. Pour cela, la fonction `OpenFolder(icon)` affiche la fenêtre et son contenu, et retourne la fenêtre créée. Une fenêtre est constituée d'une barre de titre, qui permet de déplacer la fenêtre, d'une case de fermeture, qui permet de la fermer, et des icônes qu'elle contient.

Modifier la machine à états de la question 1 pour qu'un double clic permette d'ouvrir un icône de type dossier et pour que l'on puisse déplacer et fermer les fenêtres ainsi créées.

3. Le drag-and-drop est enrichi de l'interaction suivante : lorsque l'on déplace un icône sur un icône de type dossier et que l'on reste avec le bouton de la souris enfoncé sur cet icône destination pendant au moins 0,5s, le dossier s'ouvre et permet ainsi de placer l'icône précisément à l'intérieur du dossier destination, ou bien de continuer la navigation dans les dossiers en allant à nouveau sur un icône de sous-dossier et en attendant son ouverture, etc. Cette interaction est illustrée ci-dessous :



(a) déplacement et attente    (b) animation d'ouverture    (c) fenêtre ouverte

Modifier la machine à états de la question 2 pour intégrer cette interaction.

4. Dans l'interaction de la question 3, les fenêtres s'ouvrent mais ne sont jamais refermées. Supposons que lors d'une interaction on a ouvert la fenêtre A, puis la fenêtre B à partir d'un icône de A, et enfin la fenêtre C à partir d'un icône de B. Si le curseur sort de la fenêtre C pour aller ailleurs que dans A, B ou C, on ferme les trois fenêtres A, B et C. S'il sort de C pour aller dans A, on ferme seulement B et C. S'il sort de C pour aller dans B, on ferme seulement C.

Modifier la machine à états pour prendre en compte cette fermeture automatique des fenêtres.

5. Afin de donner à l'interface iconique une plus grande réalité physique, on veut pouvoir "lancer" les icônes : lorsque l'on relâche un icône ailleurs que sur un autre icône, il poursuit sa course en fonction de la vitesse et de la direction du déplacement. On simule une force de frottement : chaque déplacement est plus court que le précédent d'un facteur réglable  $f$  ( $0 < f < 1$ ), et l'icône s'arrête si son déplacement est inférieur à 2 pixels ou s'il atteint le bord de l'écran.

Modifier la machine à états en conséquence.

**Note :** on dispose d'une fonction `GetTime()` qui retourne le temps écoulé en millisecondes depuis le dernier appel à `ResetTime()`, et d'un événement "Timeout" dont on peut programmer le déclenchement au bout de  $n$  millisecondes par la fonction `Arm(n)`.

## C. Modèle conceptuel (7 points)

On considère l'interface d'un système de courrier électronique classique :

- Une ou plusieurs boîtes aux lettres reçoivent des messages ;
- On peut créer une hiérarchie de dossiers pour y archiver les messages ;
- On peut afficher le contenu des boîtes aux lettres et des dossiers sous forme de listes ;
- On peut afficher le contenu de chaque message (on ne considère que des messages sans attachements) ;
- On dispose d'un annuaire permettant de stocker les noms et adresses des destinataires et de créer des listes de distribution ;
- On peut composer de nouveaux messages (sans attachements).

1. Identifier les objets et opérations du modèle conceptuel de cette interface.

2. L'interface retenue est organisée comme la plupart des applications actuelles : un panneau sur la gauche contient les boîtes aux lettres et dossiers, le panneau du haut liste les messages de la boîte ou du dossier sélectionné, le panneau du bas affiche le message courant. L'annuaire et les messages en cours de composition apparaissent dans des fenêtre séparées.



Donner la table fonctionnelle de cette interface, en prévoyant que le système sera utilisé aussi bien par des utilisateurs novices que des experts. Justifier les choix par l'utilisation de critères ergonomiques.

3. L'interface ci-dessus impose qu'une seule boîte ou dossier soit ouverte à la fois et que seul le message courant de cette boîte ou dossier soit affiché. Modifier l'interface afin que l'on puisse avoir plusieurs onglets dans les panneaux "liste des messages" et "message courant". Faire en sorte que les messages en cours de composition soient des onglets du panneau "message courant".

Donner la table fonctionnelle de cette nouvelle interface.

Quels problèmes d'utilisabilité anticipez-vous ?