

**Master Informatique 1ère année - Université Paris-Sud**  
**Ingénierie et Systèmes Interactifs**  
**Examen - 30 avril 2008 - 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. Expliquer la différence entre un système de fenêtrage et une librairie graphique. Expliquer comment le contenu d'une fenêtre est mis à jour par le système de fenêtrage lorsqu'elle est mise au premier plan.
2. Comparer les trois facettes d'un widget avec les composants du modèle MVC. Qu'en concluez-vous ?
3. Définir le principe ergonomique d'homogénéité et illustrer avec trois exemples.
4. Décrire les menus circulaires et donner leurs avantages et inconvénients par rapport aux menus linéaires. Comparer les "toolglases" aux boîtes de propriétés (ou inspecteurs) et aux palettes classiques.

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

**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.

**Il est demandé d'être précis dans vos réponses, notamment en expliquant comment sont traités les cas limites.**

On considère une surface tactile, comme l'iPhone d'Apple ou la table Surface de Microsoft, capable de détecter plusieurs points de contacts simultanément. Dans cet exercice, on se limite à deux points de contacts simultanés au maximum.

La surface tactile envoie des événements lorsqu'un doigt de l'utilisateur touche la surface, glisse dessus, et quitte la surface. Cependant, la surface tactile *ne sait pas quel doigt* l'utilisateur utilise, ni si les deux points de contact correspondent à deux doigts d'une même main, ni d'une même personne.

Lors du premier contact, l'évènement T1Down est envoyé, puis T1Move est envoyé chaque fois que le doigt glisse sur la surface, enfin T1Up est envoyé lorsqu'on relache ce doigt. Si, pendant qu'un premier doigt est posé, on pose un

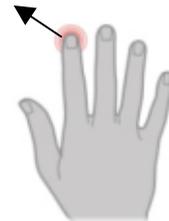
second doigt, l'événement T2Down est envoyé, puis T2Move est envoyé chaque fois que ce doigt bouge, et enfin T2Up est en envoyé lorsqu'on le relâche.

Il résulte de cette description que si l'on pose un doigt, puis un second, et que l'on relève le premier, puis le second, on a la séquence suivante d'événements (\* et l désignent respectivement l'itération et l'alternative) :

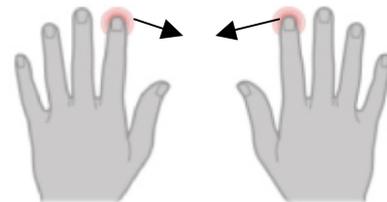
T1Down T1Move\* T2Down (T2Move|T1Move)\* T1Up T2Move\* T2Up

Chaque événement contient les coordonnées du doigt au moment de son envoi.

**1.a** Donner la machine à états d'une interaction qui permet de déplacer un objet avec un doigt. *Note* : De la même façon que les événements de la souris peuvent être suffixés par OnShape ou OnTag dans SwingStates, les événements T1Down, T1Move, etc. peuvent être suffixés par OnShape et OnTag.



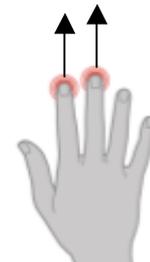
**1.b** Compléter la machine ci-dessus pour qu'elle permette de changer la taille de l'objet manipulé avec deux doigts.



Expliquer ce qui se passe si l'on relève l'un des doigts puis qu'on le repose pendant l'interaction.

*Note* : on ne fera pas les calculs de changement de taille de l'objet manipulé, on appellera simplement une méthode `obj.TouchResize(p1, p'1, p2, p'2)` avec les positions successives de chacun des deux doigts.

**2.** On utilise les interactions ci-dessus pour déplacer et changer la taille des fenêtres d'une application sur une table interactive. Pour faire défiler le contenu des fenêtres, au lieu d'utiliser une barre de défilement, on glisse les deux doigts d'une main de façon parallèle à l'intérieur de la fenêtre.



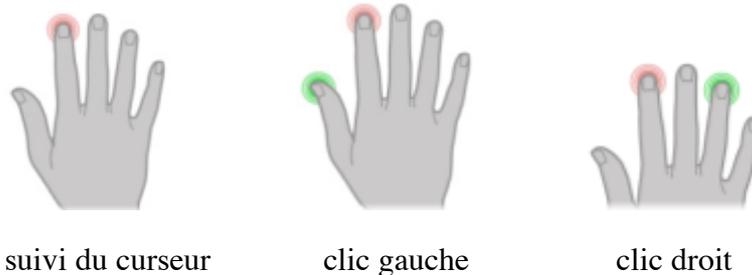
Ajouter cette interaction à la machine de la question 1, en expliquant quel est le critère utilisé pour la distinguer de l'interaction de changement de taille.

**3.** Pour pouvoir interagir avec des applications classiques sur une table interactive, il faut pouvoir simuler l'utilisation d'une souris. Pour cela il faut résoudre deux problèmes : d'une part, le curseur ne peut suivre la position d'un doigt que lorsque celui est posé sur la table ; d'autre part, il faut simuler les différents boutons de la souris.

On utilise les principes suivants pour simuler la souris :

- si l'on pose un doigt (l'index de la main droite), le curseur suit ce doigt ;
- si, pendant que l'index est posé, on pose le pouce et qu'on le relâche, cela produit l'équivalent d'un clic gauche ;
- si, pendant que l'index est posé, on pose le majeur ou l'annulaire et qu'on le relâche, cela produit l'équivalent d'un clic droit.

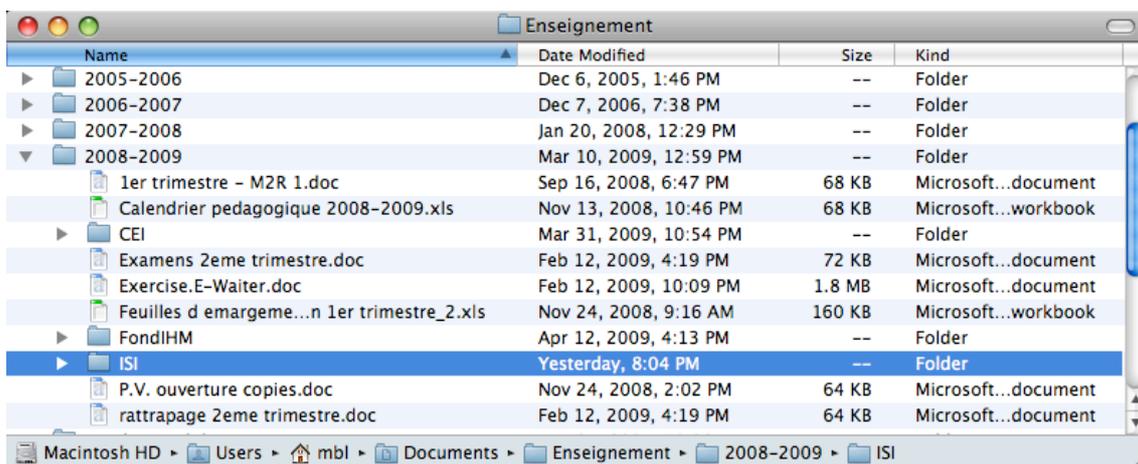
Expliquer comment vous pouvez distinguer les clics gauche et droit. Implémenter la machine à états correspondante, en détectant un clic seulement si le doigt est posé puis relâché en moins de 200ms. La machine appellera les fonctions RightClick(p) ou LeftClick(p) selon le clic qui sera détecté (p est la position du curseur au moment du clic).



4. Compléter la machine à états de la question 3 avec l'interaction de tracé (ou "drag") : si on pose l'index, puis un second doigt et qu'on bouge la main (donc les deux doigts en même temps), on démarre une interaction de tracé, jusqu'à ce qu'on relâche le second doigt. La machine appellera les fonctions StartDrag(p), Drag(p), StopDrag(p) aux différentes étapes de l'interaction.

### C. Modèle conceptuel (7 points)

On considère un explorateur de fichiers qui permet d'afficher, de naviguer et de manipuler une hiérarchie de fichiers et de dossiers. On s'intéresse uniquement à la représentation sous forme de liste, comme illustré ci-dessous : chaque fichier ou dossier est affiché sur une ligne, avec certaines de ses propriétés. Un dossier peut être fermé ou ouvert, dans ce dernier cas son contenu est également affiché.



Les opérations que l'on souhaite effectuer sur les fichiers et dossiers sont :

- Changer le nom d'un fichier ou d'un dossier ;
- Déplacer un fichier ou un dossier dans la hiérarchie ;
- Copier un fichier ou un dossier ;
- Ouvrir un fichier ou un dossier ;
- Détruire un fichier ou un dossier ;
- Créer un nouveau dossier.

L'affichage d'un dossier se fait dans une fenêtre. On peut choisir la colonne qui détermine le critère de tri pour le contenu de chaque fenêtre. Plusieurs fenêtres peuvent être ouvertes simultanément.

1. Identifier les objets et les opérations du modèle conceptuel de cette interface, en justifiant vos choix.
2. Compléter la table des objets et la table des opérations décrivant le modèle conceptuel de cette interface. Pour chaque opération, donner si possible au moins deux interactions pour effectuer l'opération.

*Table des objets*

Objets	Représentations	Propriétés	Opérations

*Table des opérations*

Opérations	Commandes	Feed-back	Réponses

3. Utiliser les critères ergonomiques de simplicité et de concision pour comparer cette interface avec une interface à ligne de commande.
4. Proposer une modification de l'interface graphique permettant de manipuler un ensemble de fichiers spécifié par une expression régulière simple, comme par exemple \*.doc.
5. Ajouter à l'interface du navigateur de fichiers la notion de "tag". Un tag est une étiquette associée à un ou plusieurs fichiers ou dossiers. Chaque fichier ou dossier peut avoir un nombre arbitraire de tags. L'interface doit notamment permettre de :
  - créer un nouveau tag ;
  - ajouter un tag à un fichier ou un dossier ;
  - enlever un tag d'un fichier ou d'un dossier ;
  - manipuler tous les fichiers et dossiers d'une fenêtre ayant un tag donné.

**Question facultative :** proposer des "tags intentionnels" définis par une expression régulière, permettant de répondre aux questions 4 et 5 de manière unifiée .