Master Informatique 1ère année - Université Paris-Sud

Ingénierie et Systèmes Interactifs

Examen - 7 mai 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. Donner la formulation de la Loi de Fitts et expliquer ce qu'elle signifie.

2. Décrire les trois services minimum du noyau fonctionnel d'une application interactive et les illustrer chacun par un exemple.

3. Expliquer le rôle de chaque composant du modèle MVC. Montrer comment la boîte à outils Java Swing met en œuvre ce modèle.

4. Faire le schéma montrant les différentes couches logicielles permettant la programmation des interfaces graphiques et donner un exemple de chaque couche.

B. Modélisation de l'interaction (9 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.

1. Donner la machine à états qui permet d'afficher des bulles d'information ("info-bulles") lorsque le curseur de la souris passe au-dessus d'un objet (par exemple un bouton ou un icone) et reste immobile pendant 500ms. L'info-bulle disparaît lorsque le curseur sort de l'objet.

On utilisera les événements *Enter* et *Leave* qui sont envoyés lorsque le curseur entre ou sort d'un objet, et l'événement *Timeout* qui est émis *n* millisecondes après l'appel à la fonction *arm(n)*. On pourra également utiliser la méthode *disarm()* qui annule l'appel à *arm* si l'événement *Timeout* n'a pas encore été émis, et qui ne fait rien sinon. Remarque : il n'y a qu'un temporisateur par machine à états, et chaque appel à *arm* réinitialise ce temporisateur.

Pour l'affichage et l'effacement des info-bulles, on utilisera les fonctions

```
b = ShowInfo(o)     // affiche la bulle b d'information sur l'objet o
HideInfo(b)         // efface la bulle b
```

2. La machine à états de la question 1 oblige l'utilisateur à attendre 500ms à chaque fois qu'il veut faire apparaître une info-bulle sur un nouvel objet. Une meilleure interaction consiste à faire apparaître l'info-bulle immédiatement si l'on vient de sortir d'un objet où l'info-bulle était affiché. Ainsi, si l'on a par exemple une rangée d'icones, on arrête le curseur sur le premier icone pour faire apparaître son info-bulle ; lorsque l'on déplace le curseur sur les autres icones, les info-bulles apparaissent instantanément jusqu'à ce que le curseur soit resté en-dehors d'un objet pendant plus de 500ms. Il faut alors à nouveau faire une pause de 500ms sur un objet pour faire apparaître son info-bulle.

Modifier la machine à états de la question 1 pour implémenter cette interaction.

3. On veut rendre les info-bulles actives : si l'utilisateur clique sur une info-bulle, celle-ci s'agrandit pour afficher plus d'information, s'il clique sur une bulle agrandie, celle-ci revient à sa taille initiale. Pour agrandir la bulle *b*, on utilise la fonction *MoreInfo(b)*, pour la réduire, *LessInfo(b)*.

Modifier la machine à états de la question 2 pour intégrer cette interaction. On considèrera que les objets qui peuvent afficher une info-bulle ont le tag « info » tandis que les info-bulles sont des objets graphiques marqués par le tag « tip ».

Quelle est la condition géométrique sur le placement des infos-bulles pour que cette interaction soit possible ?

4. On considère maintenant les icones d'un navigateur de fichiers. Donner la machine à états permettant de distinguer un clic (appui puis relâchement du bouton de la souris sans mouvement significatif dans un délai de moins de 400ms), un clic long (appui sur le bouton de la souris sans déplacement significatif pendant un délai d'au moins 400ms) et un drag-and-drop (appui sur le bouton de la souris puis déplacement significatif).

Le clic permet de sélectionner un icone, le clic long permet d'afficher un menu contextuel, et le drag-and-drop permet de déplacer les icones. Ajouter les actions nécessaires à la machine à états pour implémenter ces interactions.

Faire en sorte que cette machine à états communique avec celle de la question 3 pour que, si une info-bulle est affichéee sur un icone, celle-ci soit effacée lorsque l'on fait un clic long ou un drag-and-drop de cet icone. On rappelle que l'on peut créer de nouveaux événements et qu'une machine peut envoyer une événement à une autre machine par la méthode *sm.SendEvent(ev)*.

5. Compléter la machine à états de la question 4 pour permettre la sélection des icones par un lasso : lorsque l'on fait une interaction de tracé qui démarre sur le fond d'écran, au lieu de tracer un rectangle comme dans les interfaces actuelles, on fait une trace polygonale qui suit le tracé de la souris. Si la trace est fermée, c'est-à-dire si on recoupe le lasso avant de relâcher le bouton, tous les icones incluses dans la trace sont sélectionnées. On utilisera les fonctions ci-dessous pour gérer le lasso :

```
lasso = NewLasso()          // crée un nouveau lasso
AddLasso(lasso, pt)         // ajoute un point au lasso
```

LassoClosed(lasso)	// teste si un lasso est fermé ?
TagLasso(lasso, tag)	// met un tag sur tous les icones
	// contenus dans le lasso
ClearLasso(lasso)	// efface et détruit le lasso

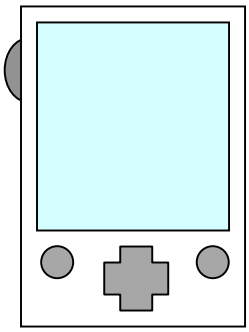
6. On veut que les lassos soient des objets persistants, similaires à des fenêtres : lorsque l'on termine le tracé d'un lasso, le polygone reste affiché, on le remplit d'un fond de couleur unie et il contient les icones qu'il entoure. On peut déplacer un lasso avec son contenu par une interaction de drag-and-drop démarrée sur le fond du lasso. On peut aussi ajouter et retirer des icones du contenu du lasso par simple drag-and-drop.

Expliquer la façon de gérer les objets graphiques représentant les lassos et les icones, en particulier leur ordre de superposition. Donner les fonctions nécessaires à la gestion des icones et des lassos, et modifier la machine à états de la question 5 pour traiter les lassos persistants.

C. Modèle conceptuel (5 points)

On considère un dispositif mobile de type téléphone portable ou PDA muni :

- d'un écran de petite taille, non tactile ;
- d'une molette que l'on peut manipuler avec le pouce ;
- d'une touche multidirectionnelle à 4 directions (N, S, E, O) ;
- de deux boutons situés juste en-dessous de l'écran ;
- d'un appareil photo dont l'objectif est au dos de l'appareil.



Vous devez concevoir l'interface de l'application de gestion de photos de cet appareil. Les objets et les opérations du modèle conceptuel sont les suivants :

- Objet *Photo* : prendre une photo, afficher une photo, envoyer par SMS
- Objet *Photothèque* : créer un album, détruire un album
- Objet *Album* : ajouter une photo, retirer une photo, lancer un diaporama

1. Définir l'aspect général de l'interface de cette application en dessinant les principales images d'écran et en expliquant son fonctionnement général. Justifier les choix avec des critères ergonomiques.

2. Donner la table fonctionnelle complète de cette interface.