Prénom :_____ **Nom de famille** :_____

Matricule :_____ **Signature** :_____

École Polytechnique de Montréal
Génie informatique

LOG2420 (Analyse et conception d'interfaces utilisateurs)
Professeur : Michel Desmarais

Examen intra, solutionnnaire – Hiver 2007
19 février 2007

Durée : 150 minutes
Pondération : 20%

La documentation et la calculatrice ne sont pas permises

L'examen comporte **10** pages et **5** questions. Le chiffre entre crochet au début de chaque question indique le nombre de points assignés à la question.

Répondre à toutes les questions.

Écrivez vos réponses sur le cahier de réponse. Vous n'avez pas à remettre le questionnaire sauf si, bien sûr, vous choisissez d'écrire certaines de vos réponses sur celui-ci.

Réservé au correcteur :

	Résultat
1 (6)	
2 (6)	
3 (0)	
4 (2)	
5 (0)	
Total (20)	

1. L'annexe A reproduit quelques fenêtres de l'application Mapedit que nous avons vue en classe. De plus, une animation est affichée au projecteur pour la tâche de redimentionner une région active. Elle démontre que la tâche peut s'effectuer à la fois par le menu ("move") et par un bouton.

   (a) **[3 pts.]** Effectuez une inspection cognitive de la tâche de redimensionner une région active pour l'application Mapedit. Limitez votre réponse à une page en portant votre analyse sur les points qui vous semblent les plus importants. Une page est insuffisante pour faire une analyse complète, mais il faut démontrer que vous savez comment l'effectuer.

   Solution : Il faut décomposer la tâche en une séquence d'actions que l'utilisateur doit effectuer et identifier :

   - quelles sont les connaissances et informations requises;
   - les sources susceptibles d'imposer une forte charge cognitive;
   - les erreurs potentielles et
   - se référer aux quatres questions clés pour déterminer les accrocs potentiels

   Nous ne relevons ici que les points d'intérêt et omettons ceux dont il n'y a rien d'important à relever.

   - Action : activer le mode de déplacement/redimensionnement
     - Connaissances
       - il y a des modes dans l'application ;
       - le texte d'explication au bas apparaît lorsqu'on clique sur une icone;
       - la redimension d'une région peut se faire selon des ajouts ou des retraits de points de contour, ou par le déplacement de points de contour; l'exemple ici porte sur le déplacement.
       - la fonction pour redimentionner une région se trouve sous le menu Tools et se nomme "move"
       - alternative au menu : l'icone avec une sphère rose
     - Charge cognitive
       - Déduire que l'application fonctionne avec des modes correspondants aux items du menu Tools, aux textes du bas et aux icones.
       - Déduire dans quel mode on est, ce qui implique soit de s'en souvenir, soit de lire le texte en bas (qui dépend du mode), soit l'identifier par les icones (un très léger effet de bouton pressé).
       - Décider si le redimensionnement doit se faire par ajout/retrait de points ou par déplacement de points.

    * Faire le choix de l'icone ou de l'item du menu Tools approprié. La première fois, ceci nécessite de faire la correspondance entre les items de menu et les icones et de lire le texte qui apparaît pour expliquer l'icone. Par la suite, il faut mémoriser les fonctions des icones pour éviter la lecture.
    * Balayer l'image pour trouver les régions actives entourées d'un cadre formé d'une ligne noire.
  - Erreurs potentielles
    * La compréhension des icones n'est pas évidente et prête à erreur, mais on peut généralement rapidement réaliser qu'un mauvais choix est fait et récupérer facilement (on pourrait analyser en détail des scénarios d'erreurs).
  - Questions
    * La question de propension à effectuer la bonne action pour un nouvel utilisateur pose problème s'il ne comprend pas qu'il doit être dans le mode correspondant. Il est toutefois probable qu'il comprenne mieux avec les menus et une fois qu'il fera le lien entre les icones et le texte du bas, et le mode indiqué par des boutons pressoir dans le menu Tools.
- Action : cliquer sur la région active à vérifier
  - Connaissances
    * Les régions actives sont indiquées par les rectangles noirs.
    * Il faut premièrement cliquer une région pour activer la fonction de redimensionnement.
    * La croix au milieu de la région permet le déplacement et les croix sur les contours du rectangle permettent de déplacer contour.
  - Charge cognitive
    * Il n'y a pas de charge cognitive élevée pour cette étape et c'est d'ailleurs la qualité de cette application de faciliter l'identification des régions actives.
- Action : déplacer un des coins pour redimensionnner
  i. Connaissances
    A. Les croix aux coins du rectangle sont les points que l'on peut déplacer.

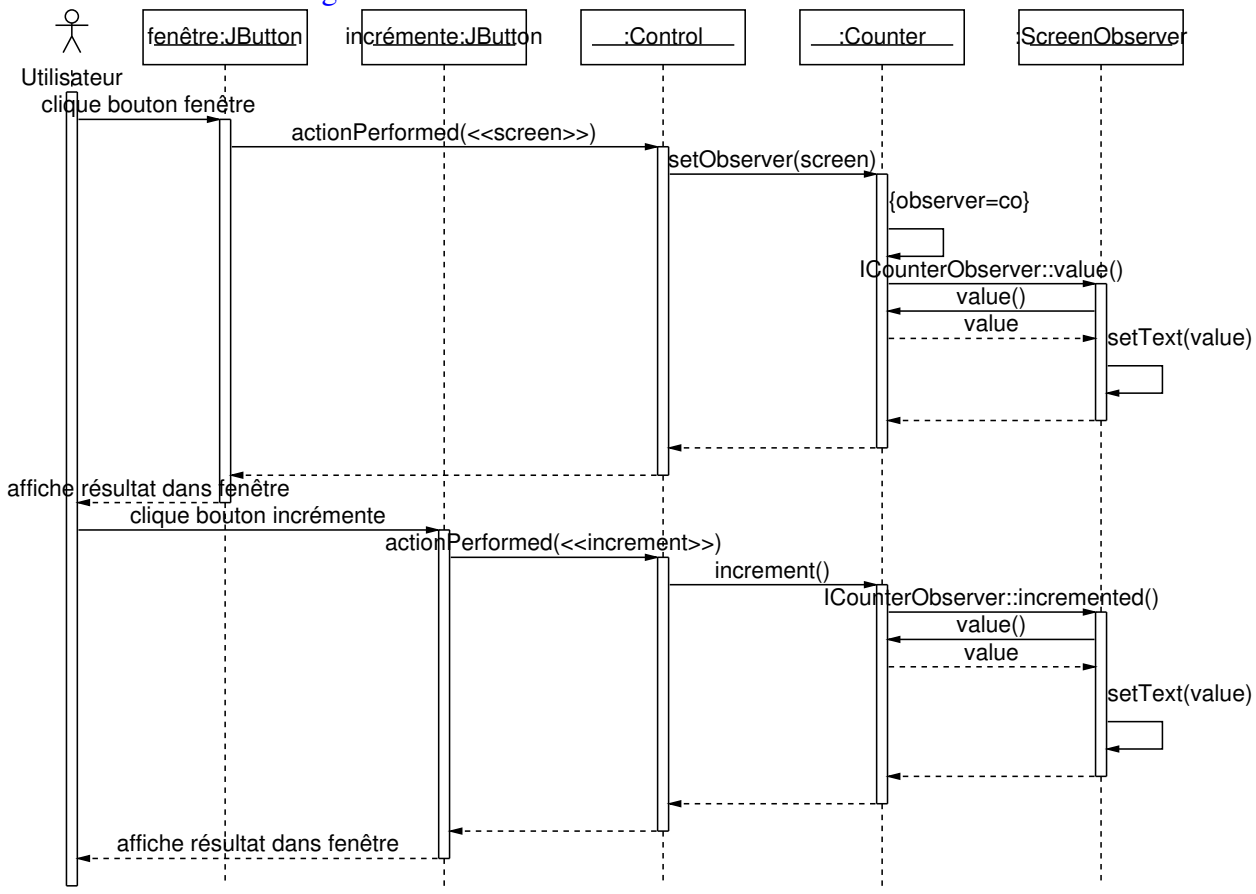
(b) **[3 pts.]** Suggérez les **trois améliorations** à l'interface qui vous semblent les plus pertinentes. Vous avez la liberté de modifier la fonctionnalité si vous jugez que c'est nécessaire. Au besoin, illustrez vos changements sur les images de l'annexe A et/ou redessinez schématiquement l'interface envisagée.

2. L'annexe B reproduit le code de l'application `CounterDemo` vu en classe. Notez que ce n'est pas le code complet de l'application.

(a) **[3 pts.]** Expliquez ce qui se produit si l'utilisateur appuie premièrement sur le bouton "Fenêtre", puis sur "Incrémente". Expliquez notamment la séquence d'appels de méthodes des différents objets puis la gestion des événements. Limitez votre explication à 1 page.

Solution :

Le CounterDemo est un exemple de modèle MVC basé sur le patron *observateur* et où la classe `Counter` est le modèle, les classes qui implémentent `CounterObserver` sont les vues (`ScreenObserver` et `ConsoleObserver`) `Control` est le contrôleur. La première action a pour effet d'affecter la vue `ScreenObserver` comme observateur du modèle, `Counter` et d'afficher sa valeur. La seconde action incrémente le compteur du modèle et l'affiche dans la vue active. Le détail des appels est affiché dans le diagramme de séquence suivant. Noter que nous simplifions en présumant que les boutons transmettent directement les événements plutôt qu'à travers le gestionnaire d'événements.



(b) **[3 pts.]** Complétez le tableau ici-bas en associant à chaque rôle/service les deux éléments suivants :

i. un, et un seul des trois composants de l'architecture MVC ((1) modèle, (2) vue et (3) contrôleur) le service;

ii. l'objet/interface, ou les objets de l'application qui remplissent, partiellement ou entièrement, le rôle/service. Les objets/interfaces à considérer sont `Counter`, `Control`, `CounterObserver`, `ConsoleObserver`

No	Rôle/Service	M, V ou C (du MVC)	Objet(s) java
1	Encapsule l'état de l'application	M	Counter
2	Effectue le rendu du modèle	V	ScreenObserver
3	Expose la fonctionnalité de l'application	M	Counter
4	Sélectionne la vue	C	Control
5	Transmet au contrôleur les actions de l'utilisateur	V	ScreenObserver
6	Établit la correspondance des actions utilisateur à la mise à jour du modèle	C	Control

3. **[2 pts.]** Décrivez les différences et les points communs entre un cas d'utilisation et une tâche (1/2 page).

   Solution :

   - les deux servent à décrire les besoins fonctionnels pour le développement d'une application;
   - la tâche est décrite plus tôt dans le processus, lors de l'analyse des besoins; elle est indépendante de la description d'un système ou d'une interface; elle permet notamment de définir ce qui peut/doit ou non être automatisé;
   - on associe souvent à l'analyse de la tâche des données de fréquence, de dépendances et d'imbrication;
   - les cas d'utilisation nécessitent une délimitation de ce qui est automatisé et des fonctions que l'on attend du système; ils arrivent plus tard dans le processus, lors de la définition des exigences et donnent une idée plus précise de l'interaction.

4. **[2 pts.]** Expliquez l'effet de l'apprentissage sur l'utilisation d'une interface, sachant que l'apprentissage comprend le domaine d'application, l'environnement informatique, et l'apprentissage de l'interface de application elle-même. Référez-vous autant que possible aux notions théoriques du cours, notamment à l'impact de l'apprentissage sur la charge cognitive, le phénomène de *patrons* (*chunks*), la loi de la pratique, etc.

Solution : L'apprentissage est un cumul de connaissances, dont certaines sont des patrons, qui nous permet d'effectuer des tâches plus efficacement. La loi de la pratique décrit l'évolution du temps pour effectuer une tâche en fonction du nombre d'essais précédents et comporte deux deux constantes, une qui décrit la rapidité à laquelle on s'améliore et l'autre qui est basé sur le temps pour effectuer le, ou les quelques premiers essais.

La théorie des patrons nous démontre qu'au long de l'apprentissage, l'individu perçoit un problème ou une interface de façon très différente, selon qu'il reconnaît des situations eet qu'il sait quoi faire. Les patrons peuvent diminuer considérablement la charge cognitive lié à une tâche en filtrant l'information à analyser et réduisant le nombre de "chunk" à mémoriser.

5. **[4 pts.]** Vous êtes un analyste dont le nom commence à être reconnu chez Apple. Nous sommes à l'époque de la conception du iPod et de ses applications. Vous êtes responsable de la conception d'une application de gestion des fichiers de musiques (le iTunes). Le Vice-Président vous a déjà expliqué comment cette application est critique pour le iPod et pour la compagnie. Il vous a répété plusieurs fois qu'il est important de repenser ce qui existe déjà comme applications en termes des vrais besoins des utilisateurs. L'application doit être reconnue comme la première d'une nouvelle génération de ces applications. Il ne veut pas qu'un simple lecteur de fichiers de musique avec une fonction additionnelle de permettre la copie fichiers sur le iPod.

   Il vous demande donc de lui fournir, en une page ou deux (rappelez-vous, c'est un VP et il n'aime pas entrer dans les détails!), un plan d'analyse et de de conception de l'interface, incluant le cycle de développement et les activités que vous allez entamer dans les prochains mois pour vous assurer de livrer une application selon, comme il le dit, les "vrais besoins des utilisateurs". Soyez aussi spécifique que possible et ne vous contentez pas simplement de faire une liste de certaines des activités de ISO13407. Justifiez votre réponse pour que le VP soit en confiance et comprenne que vous savez ce qu'il faut faire!

   (barème : 2 pts pour le plan, 2 pts pour la justification)

   Solution : La prochaine génération du gestionnaire de musique personnel

   Les applications actuelles de musique sont mono-fonctions. Elle ne traitent qu'un des aspects du processus de découverte, d'achat, d'organisation et d'écoute de la musique. L'application envisagée regroupera ces fonctions en un tout cohérent et soutiendra toutes ces tâches de manière intégrée, de la découverte, à l'achat, à l'organisation et à l'écoute de la musique. l'échéancier est extrèmement serré et il est possible que la version actuelle ne couvre pas toute la gamme de services envisagés, mais la conception de cette première version sera à tout le moins une démonstration convaincante de cette vision de l'application de seconde génération.

   Un plan général des activités engagées suit. Les chiffres indique les semaines.

Déterminer le contexte d'utilisation

+

[1–4] Il faut établir quel sera le contexte d'utilisation, notamment les caractéristiques de l'appareil IPod, les utilisateurs visés, le nombre de chansons, les façons de récupérer la musique (le site de vente de musique de Apple)
La direction et les responsables marketing doivent être consultés et confirmer le contexte d'utilisation. Une série d'activités sont impliquées pour établir ce contexte et c'est pourquoi l'activité s'échelonne sur au moins 4 semaines.

[1] **Objectifs d'entreprise** . Il faut bien cerner les objectifs stratégiques de l'application (comment elle se situe par rapport à la compétition, les produits connexes (sites de musique) quel est le marché visé, l'envergure, etc.).
La direction et les responsables marketing doivent être consultés et confirmer ces objectifs.

[2–4] **Groupes de discussion**. Deux à trois groupes de discussion seront organisés afin de recueillir des commantaires de différents intervenants, notamment des utilisateurs représentatifs, des gens impliqués dans le développement matériel du Ipod, des distributeurs et vendeurs, des gens du marketing impliqués dans la publicité et les analystes.

[3–5] **Études des produits concurrents et connexes** (ex. les sites de musique, les "CD ripper")

[8] **Sondage pour établir le profil des utilisateurs**. Quelques informations à recueillir que l'on peut déjà envisager :

- Profil socio-économique
- Lecteurs et autres appareils connexes achetés et utilisés
- Musique écoutée: fréquence d'écoute, genre, endroit, matériel
- Musique achetée ou obtenue d'une autre façon: quels magasins ou sites? Comment trouvent-ils ce qu'ils cherchent? Comment découvrent-ils de nouveaux artistes ou pièces d'intérêt?
- Organisation de la musique: comment les utilisateurs regroupe-ils leur musique et retrouvent-ils leur artistes/chansons?

Exigences et spécifications

+

[10] **Document d'exigences et de tâches**. Une première ébauche des les exigences générales sera produite à la 10ième semaine. Les grandes tâches (fonction) qui devront être effectuables avec l'application seront décrites et étayées la forme de scénarios d'utilisation.

Conception et évaluation en itération

Quelques itérations de conception et d'évaluation des solutions nous amèneront vers le développement de la version opérationnelle de l'application. Les exigences et spécifications seront révisées en conséquence au long du processus itératif.

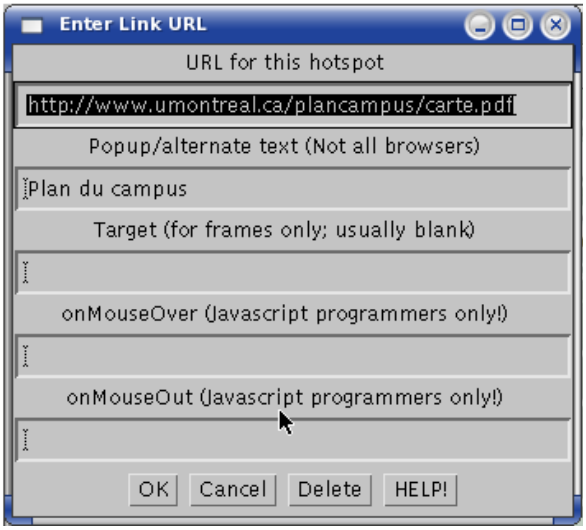
+

[10–20] **Maquettes de concepts**. Au moins deux concepts d'applications seront élaborés. Les maquettes feront l'objet d'évaluation heuristiques et d'inspection cognitive et elles seront éventuellement testées auprès d'une vingtaine d'utilisateurs potentiels. De ces maquettes semi-fonctionnelles, un ou deux prototypes seront développés (dépend du budget disponible).

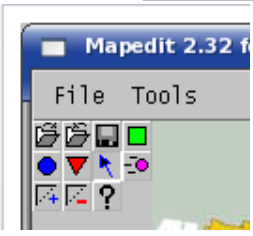
[15–25] **Prototypes fonctionnels**. Un à deux prototypes fonctionnels seront développés (basé sur les maquettes), c'est selon le budget et le coscensus autour maquettes. Les prototypes feront l'objet de tests utilisateurs.

[25–40] **Version opérationnelle**. Une fois le choix et la validation du prototype réalisés, une dizaines de semaines est prévue pour le développement et les tests de la version opérationnelle. Outre les étapes habituelles de développement, deux séries de tests utilisateurs sont prévues.

Annexe A


Enter Link URL
URL for this hotspot
http://www.umontreal.ca/plancampus/carte.pdf
Popup/alternate text (Not all browsers)
Plan du campus
Target (for frames only; usually blank)
onMouseOver (Javascript programmers only!)
onMouseOut (Javascript programmers only!)
OK
Cancel
Delete
HELP!


Mapedit 2.32 for Java
File
Tools
Help
Rectangle
Circle
Polygon
Test+Edit
Move
Add Points
Remove Points
Carte détaillée en
The Move tool allows you to move entire hotspots as well as individual points. First, click on a hotspot to select it. Then, click on and drag one of the highlighted points with the mouse. The point at the center of a hotspot is used to
Scroller


Mapedit 2.32 f
File
Tools


The Test/Edit tool allows you to test your hotspots. When you click on a hotspot, Mapedit will highlight it and display the URL window. You can use the URL window to change URLs, as well as to delete unwanted hotspots.
Scroller

Annexe B, code CounterDemo.java

```
public class Counter {
   int value = 0;
   CounterObserver observer;

   public void setObserver(CounterObserver co) {
         observer = co;
         co.value();
   }
   public void increment() {
         value++;
         observer.incremented();
   }
   public void decrement() {
         value−−;
         observer.decremented();
   }
   public int value() { return value; }
}
```

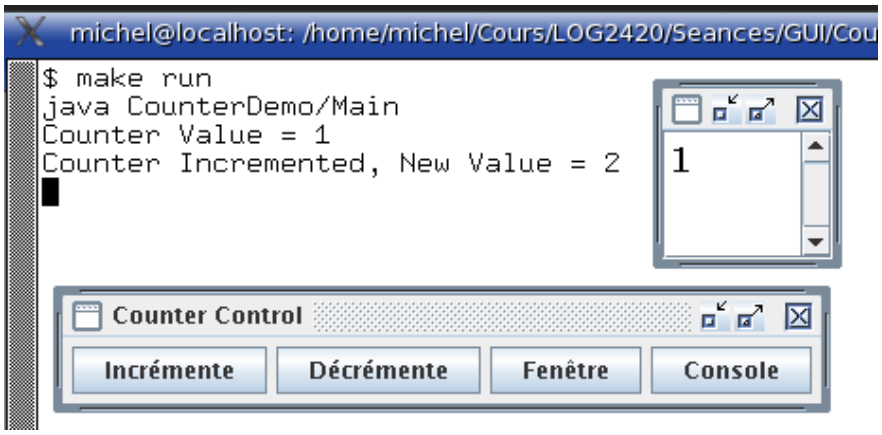
```
public class Control extends JPanel implements
ActionListener {
      protected JButton b1, b2, b3, b4;
      CounterObserver screen, console;
      protected Counter counter;
      Worker worker;

      public void initialize() {
            b1 = new JButton("Incrémente");
            b1.setActionCommand("increment");
            b1.addActionListener(this);
            b1.setToolTipText("Incrémente le
compteur");

            add(b1);
      }

      public void actionPerformed(ActionEvent e) {
            if
("increment".equals(e.getActionCommand())) {
            counter.increment();
            } else if
("decrement".equals(e.getActionCommand())) {
            counter.decrement();
            } else if
("screen".equals(e.getActionCommand())) {
            counter.setObserver(screen);
            } else if
("console".equals(e.getActionCommand())) {
            counter.setObserver(console);
            }
      }
}

      }
}
```

```
interface CounterObserver {
      public void value();
      public void incremented();
      public void decremented();
}
```

```
class ScreenObserver implements CounterObserver
{
      Screen screen;
      Counter counter;
      public void value() {
            String text =
Integer.toString(counter.value());
            screen.setText(text);
      }
      public void incremented() {
            String text =
Integer.toString(counter.value());
            screen.setText(text);
      }
      public void decremented() {
            String text =
Integer.toString(counter.value());
            screen.setText(text);
      }
```

```
}
class ConsoleObserver implements
CounterObserver {
    Counter counter;
    ConsoleObserver(Counter c) {
        counter = c;
    }
    public void value() {
        System.out.print("Counter Value =
");
        System.out.print(counter.value());
        System.out.print("\n");
    }
    public void incremented() {
        System.out.print("Counter
Incremented, New Value = ");
        System.out.print(counter.value());
        System.out.print("\n");
    }
    public void decremented() {
        System.out.print("Counter
Decremented, New Value = ");
        System.out.print(counter.value());
        System.out.print("\n");
    }
}
```

Affichage de l'interface de l'application CounterDemo :



Fin de l'examen