

POOA

TP Patrons de conception

Arnaud Lewandowski
lewandowski@lisc.univ-littoral.fr

But du TP

Il s'agit de reprendre une application existante et de la faire évoluer en utilisant au maximum les designs patterns.

Contexte et objectif

On se place dans le cadre où un client vous demande de faire évoluer son application. Bien entendu, les besoins du client évoluent eux aussi au fil du projet... Voilà pourquoi le sujet est découpé en plusieurs étapes, correspondant aux besoins exprimés au fur et à mesure par le client. Le développement itératif suivra donc cet ordre chronologique.

Ceci implique donc que le code sera amené à changer, à s'étoffer au fur et à mesure des versions. Il faudra donc s'efforcer de :

- concevoir et de coder avec **simplicité**,
- se **focaliser uniquement sur les besoins** exprimés par le client à chaque étape,
- **pratiquer régulièrement le refactoring** pour restructurer le code, le simplifier, le nettoyer, décomposer les grandes fonctions en fonctions plus petites, factoriser le code commun, etc.

Le code sera évalué en fonction de ces critères.

Chaque étape donnera lieu à une version « stable » du logiciel, que vous enverrez par mail à l'adresse indiquée ci-dessus.

Travail à faire

1 – Analyse du code

1. Analyser le code de l'application Paint fournie. Utiliser un outil de rétro-ingénierie pour construire le diagramme UML des classes de l'application.
2. À partir de ce modèle, pouvez-vous identifier les design patterns qui ont été utilisés ?

2 – Première évolution

On souhaite ajouter à l'application une barre de statut (zone horizontale, en bas de la fenêtre), en vue d'y afficher différentes informations sur le dessin. En particulier, on veut que soit affiché dans cette zone le nombre de formes graphiques dessinées. Déterminez quel patron de conception utiliser et implémentez la fonctionnalité.

3 – Deuxième évolution

On souhaite ajouter à l'application une fonctionnalité de groupement de formes. Un groupe sera composé de formes ou d'autres groupes et sera déplaçable. On implémentera aussi la fonction inverse consistant à dissocier un groupe de formes.

4 – Troisième évolution

Ajouter une fonctionnalité permettant de dupliquer une forme sélectionnée (ou un groupe de formes).

5 – Quatrième évolution

L'application propose à présent un certain nombre d'actions. On souhaite pouvoir gérer le Undo/Redo (Annuler/Rétablir). Pour cela, reprenez le code de l'application pour appliquer le patron Command aux différentes actions disponibles. Ceci vous permettra ensuite de mettre en œuvre le code nécessaire pour :

- retenir la liste des commandes exécutées au fur et à mesure de l'utilisation de l'application (création de forme, déplacement, etc.) ;
- annuler ou refaire ces commandes.

6 – Cinquième évolution

On veut pouvoir rajouter du texte sur la forme sélectionnée.

7 – Sixième évolution

La barre de statut pourrait être mieux exploitée. Faites en sorte qu'elle puisse afficher, en plus du nombre total de formes sur le dessin, des informations sur la forme (ou le groupe de formes) sélectionnée (si c'est un cercle, un rectangle, etc. ; ou dans le cas d'un groupe, combien de formes il contient, etc.).

8 – Septième évolution

Implémenter la sauvegarde d'un dessin dans un fichier au format XML, ainsi que le chargement d'un fichier XML.

9 – Huitième évolution

Ajouter la possibilité d'ouvrir un document en « lecture seule ». Indice : vous pourrez utiliser le patron Proxy.

Ressources

- La référence en ligne :
<http://www.vincehuston.org/dp>
- « Refactoring to patterns catalog » :
<http://www.industriallogic.com/xp/refactoring/catalog.html>