

OMD-S7: TP2

A rendre pour le 28/10/2024

Responsable: FEUILLATRE Hélène

Sommaire

1. Intro	3
2. V1	3
2.1. Diagramme de cas d'utilisation	3
2.2. Diagramme de classe	4
2.3. Diagramme de séquence	4
2.3.1. Couper la sélection	4
2.3.2. Entrer du texte	5
3. V2	5
3.1. Diagramme de cas d'utilisation	5
3.2. Design pattern	5
3.3. Diagramme de classe	5
4. Conclusion	5

1. Intro

faire (max 5 lignes)

2. V1

Dans la première version, les fonctionnalités présentes sont :

- La sélection du texte
- L'écriture de texte
- L'effacement du texte
- Le déplacement du curseur
- La copie de la sélection (sauvegardée dans le presse-papier)
- Le coupage de la sélection (sauvegardée dans le presse-papier)
- Le collage de la sélection qui fut au préalable stockée dans le presse-papier. Il y a deux cas possibles selon si la sélection en cours est vide ou non :
 - Si vide : cette fonctionnalité insère le contenu du presse-papier à l'endroit où le curseur se situe. présent dans le presse-papier
 - Si non vide : cette fonctionnalité remplace la sélection en cours par le contenu

2.1. Diagramme de cas d'utilisation

Au vu des fonctionnalités que doit proposer la première version de l'éditeur de texte, voici un diagramme de cas d'utilisation envisageable :

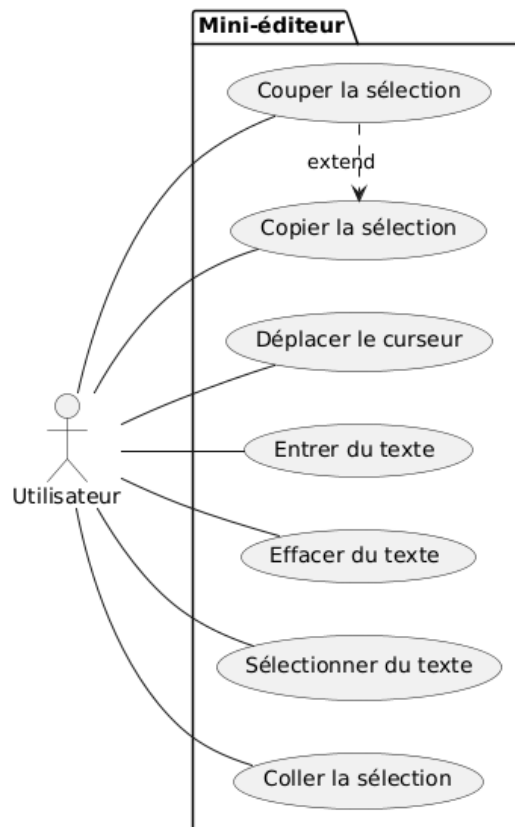


Figure 1: Diagramme de cas d'utilisation V1

2.2. Diagramme de classe

Au vu du problème que nous avons à modéliser, après avoir consulté le [catalogue](#) des design pattern présent sur [Refactoring GURU](#), nous avons décidé de nous baser sur le design pattern nommé Command.

Ce design pattern est de type comportemental, c'est-à-dire qu'ils concernent les algorithmes et l'attribution des responsabilités entre les objets. Plus précisément, le design-pattern Command permet de transformer une requête en un objet qui contient toutes les informations relatives à la requête. Cette transformation est utile car elle permet de transmettre des requêtes en tant qu'arguments de méthode, de retarder ou de mettre en file d'attente l'exécution d'une requête et de prendre en charge des opérations annulables. C'est exactement le type de structure que l'on recherche pour notre mini-éditeur.

Voici la structure exposée par ce design pattern :

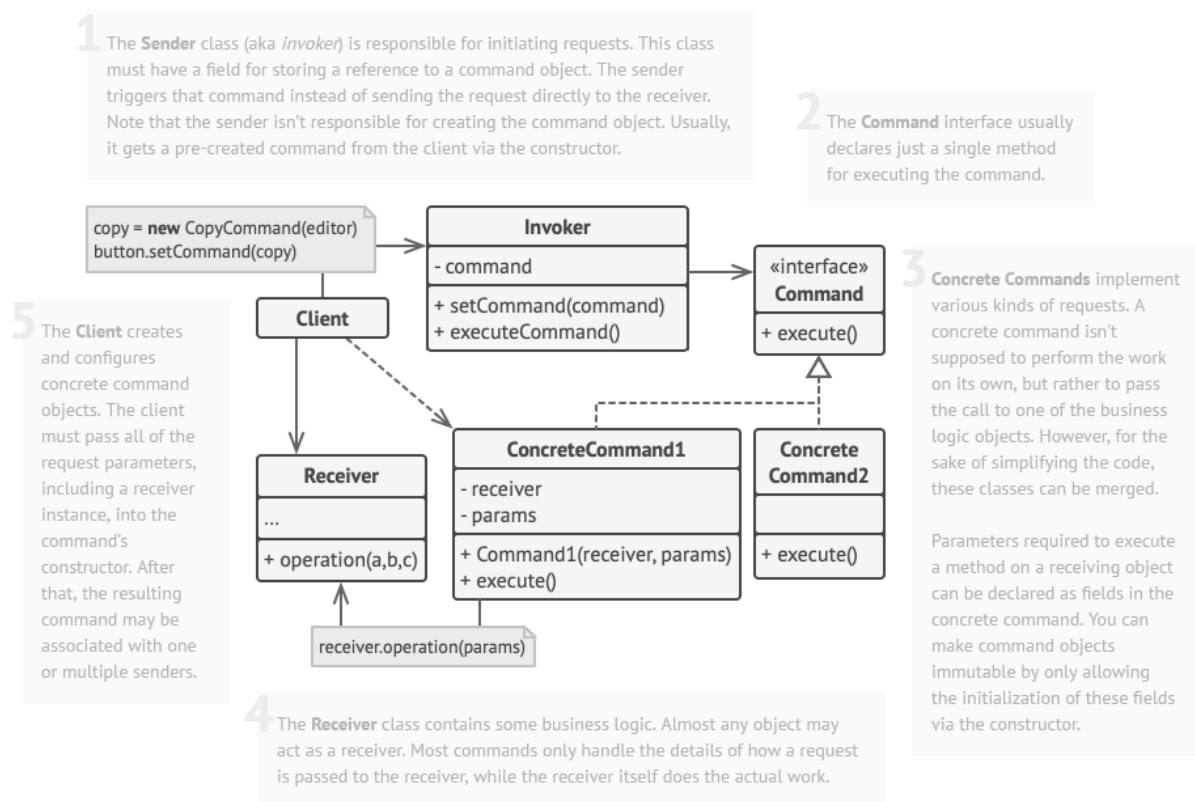


Figure 2: Diagramme de cas d'utilisation V1

Cette structure est très générale. Nous devons l'adapter à notre problème. Pour ce faire, nous devons comprendre la structure ci-dessus. Une fois cela fait, nous pourrons la customiser pour produire notre diagramme de classe.

Dans le cas de notre problème, nous avons :

- ...

2.3. Diagramme de séquence

pour bien voir comment ça marche avec les interactions avec les classes

2.3.1. Couper la sélection

2.3.2. Entrer du texte

3. V2

3.1. Diagramme de cas d'utilisation

3.2. Design pattern

<https://refactoring.guru/design-patterns> <https://refactoring.guru/design-patterns/catalog>

<https://refactoring.guru/design-patterns/memento>

3.3. Diagramme de classe

4. Conclusion

faire (max 10 lignes)