

Un projet d'application des notions et  
bonnes pratiques de développement

# Mini - éditeur

Architecture

Eddy KOKOYE & Kadidiatou SANOGO

---

## Présentation du projet :

Mini-éditeur est une application d'édition de texte avec des fonctionnalités de base. L'objectif du projet est de mettre en pratique les notions abordées en cours et TDs pour développer ce mini éditeur par l'application de design pattern.

Pour faciliter le développement, l'application a été découpé en trois (3) versions :

- La version 1 : Fournie les fonctionnalités de base que sont l'insertion, la copie, le déplacement, la suppression et la sélection de texte.
- La version 2 : est une extension de la version 1 avec pour complément de fonctionnalité l'enregistrement et le rejeu des actions de l'utilisateur.
- La version 3 : est une suite de la version 2 avec ajout des fonctionnalités défaire(undo) et refaire(redo).

## Architecture de différentes versions :

Organisation des packages de la Version 1 : Dans cette version nous avons eu à choisir le design pattern Command, afin de séparer totalement la partie affichage de la partie traitement. Ainsi nous avons créé une classe Buffer avec un attribut *Content: String* plutôt que de représenter le Buffer par un attribut String pour faciliter l'extension du Buffer.

Pour plus de clarté nous avons fait un développement par package dans chaque version.

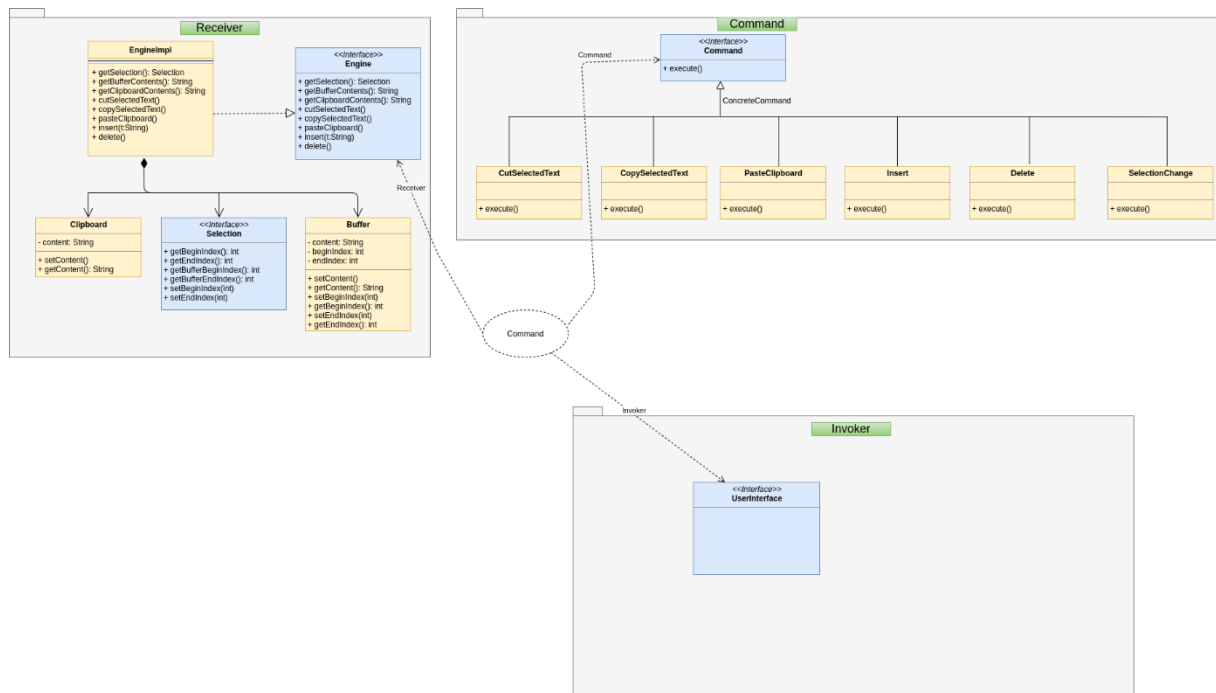
***Package Receiver*** : ce package regroupe les classes qui gèrent le Buffer

***Package ConcretCommand*** : regroupe toutes les classes de nos concrètes commandes.

***Package Test*** : réuni nos classes de tests

***Package Invoker*** : regroupe les classes qui permettent à l'utilisateur d'exécuter des commandes.

## Diagramme de Classes Version 1



## Quelques diagrammes de séquences

Diagramme de séquence Insert

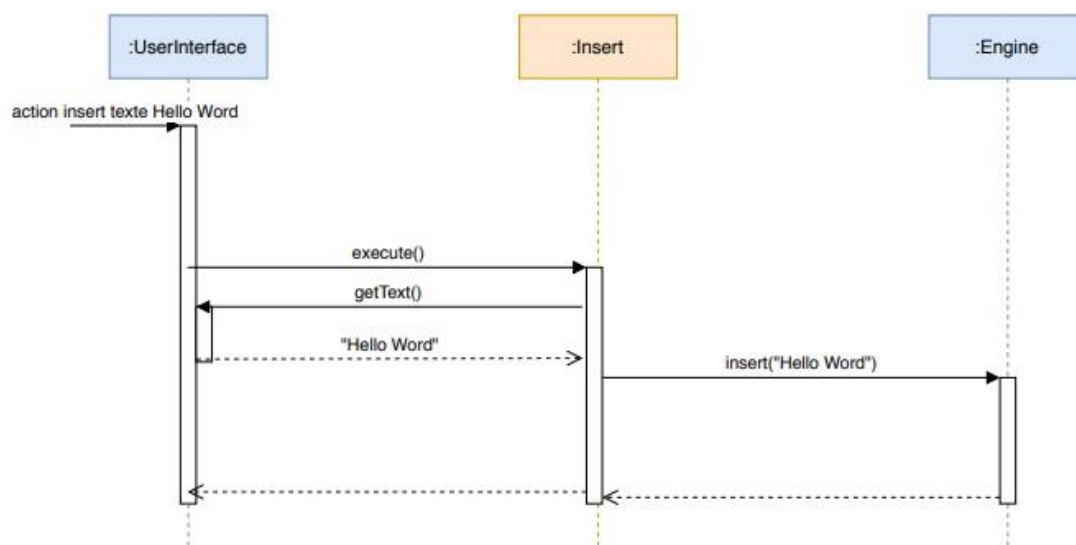


Diagramme de séquence de CutSelectedText

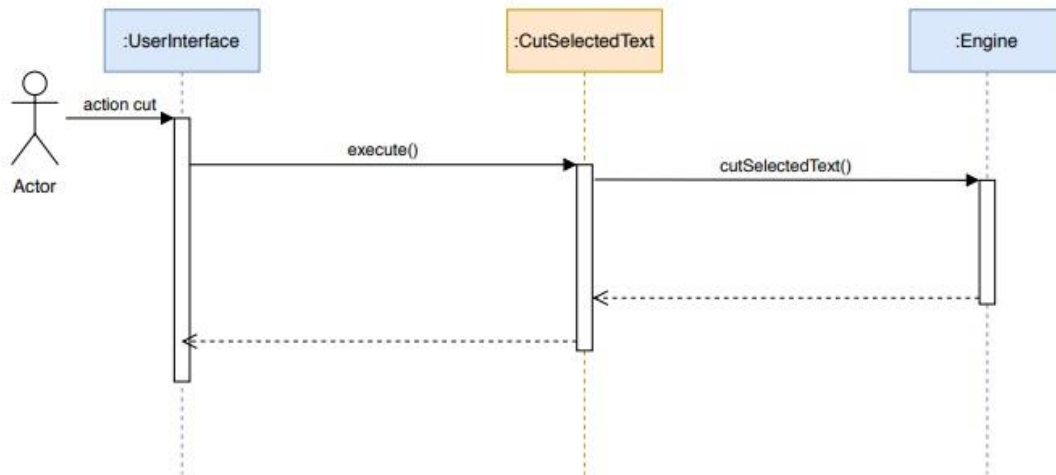
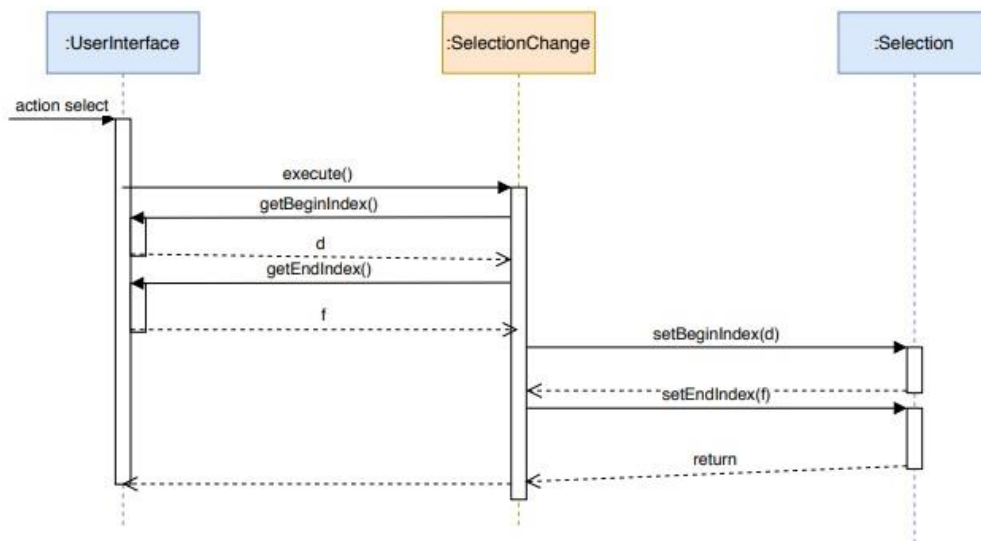
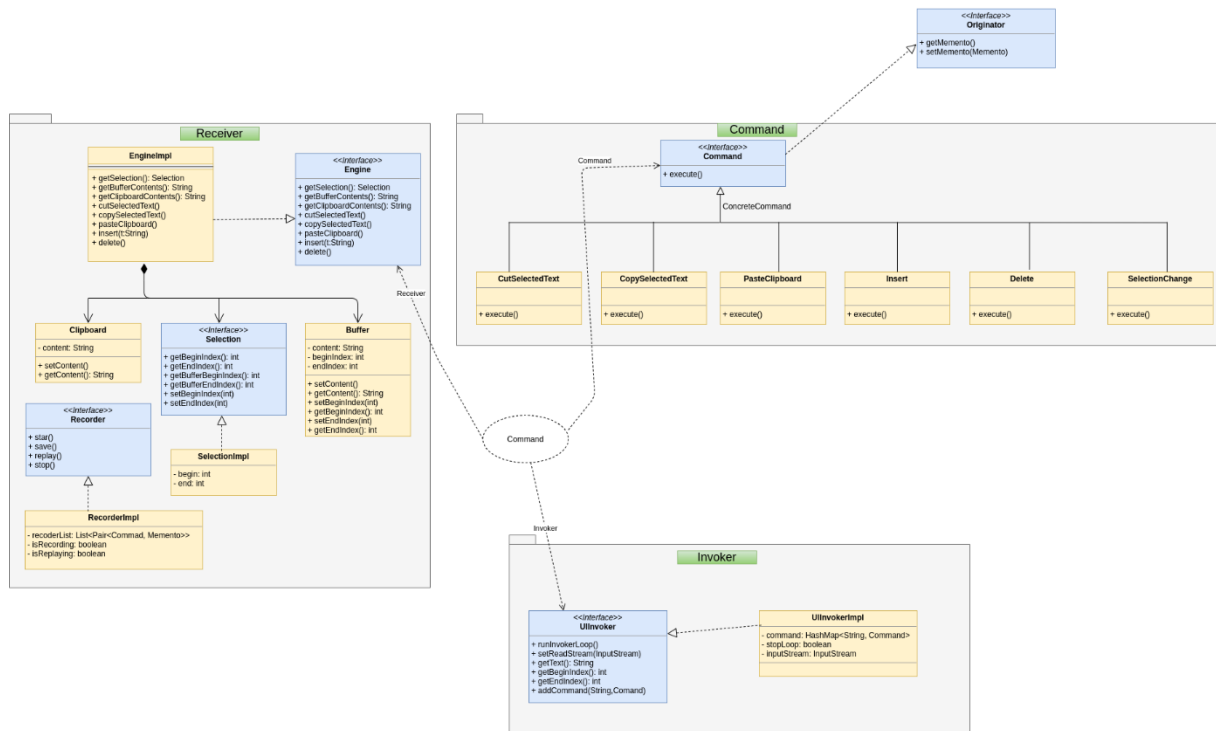


Diagramme de séquence SelectionChange



## Diagramme de classe Version 2 :

MiniEditor V2



## Exécution :

Chaque version peut être lancée séparément en téléchargeant les différentes versions sur le dépôt GitLab. Lancer la classe main de chaque version pour avoir accès à la GUI. La syntaxe des commandes que l'utilisateur doit entrer pour effectuer les différentes actions :

- 1- Insert(i) : Enter « i » pour l'insertion d'un texte dans l'Engine.
- 2- Delete(d) : Enter « d » pour supprimer la sélection.
- 3- Copier(c) : Après la sélection faire « c » pour copier la sélection.
- 4- Couper(x) : Après la sélection faire « x » pour couper la sélection.
- 5- Paste(v) : Enter « v » pour coller la sélection dans l'Engine.
- 6- Selection(select) : Enter « select » après avoir renseigné l'index de début et l'index de fin pour faire une sélection.
- 7- Start recording(start) : Faire « start » pour démarrer l'enregistrement des actions.

- 8- Stop recording(stop) : Faire « stop » après avoir démarré un enregistrement pour ainsi l'arrêter.
- 9- Replay(replay) : Faire « repaly » pour rejouer les actions enregistrées.
- 10-Undo(undo) : Entrer « undo » pour defaire une action.
- 11-Redo(redo) : Enter « redo » pour refaire une action.
- 12-Exit(e) : Faire « e » pour quitter le mini-éditeur