Diagramme de classe

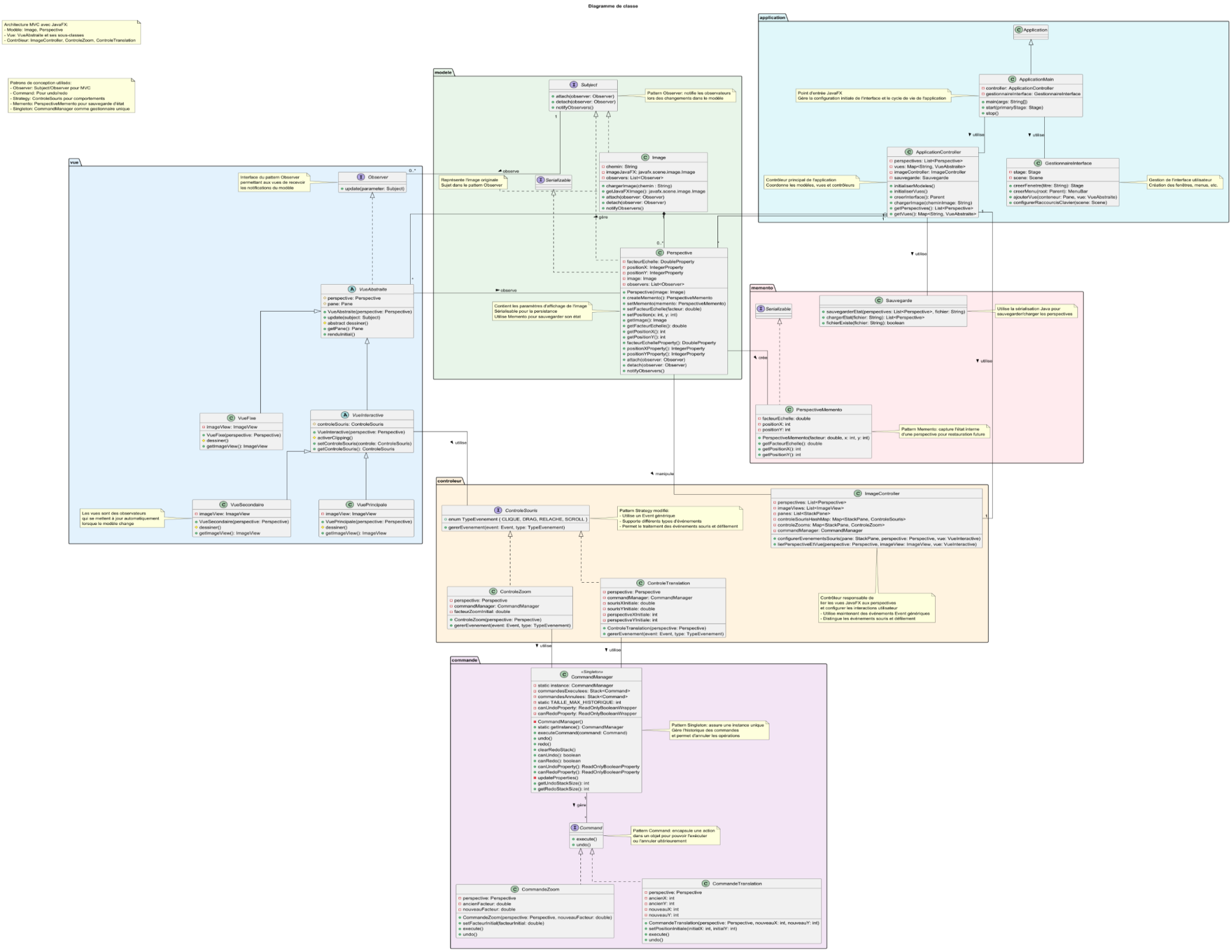

Diagramme de séquence de la commande zoom

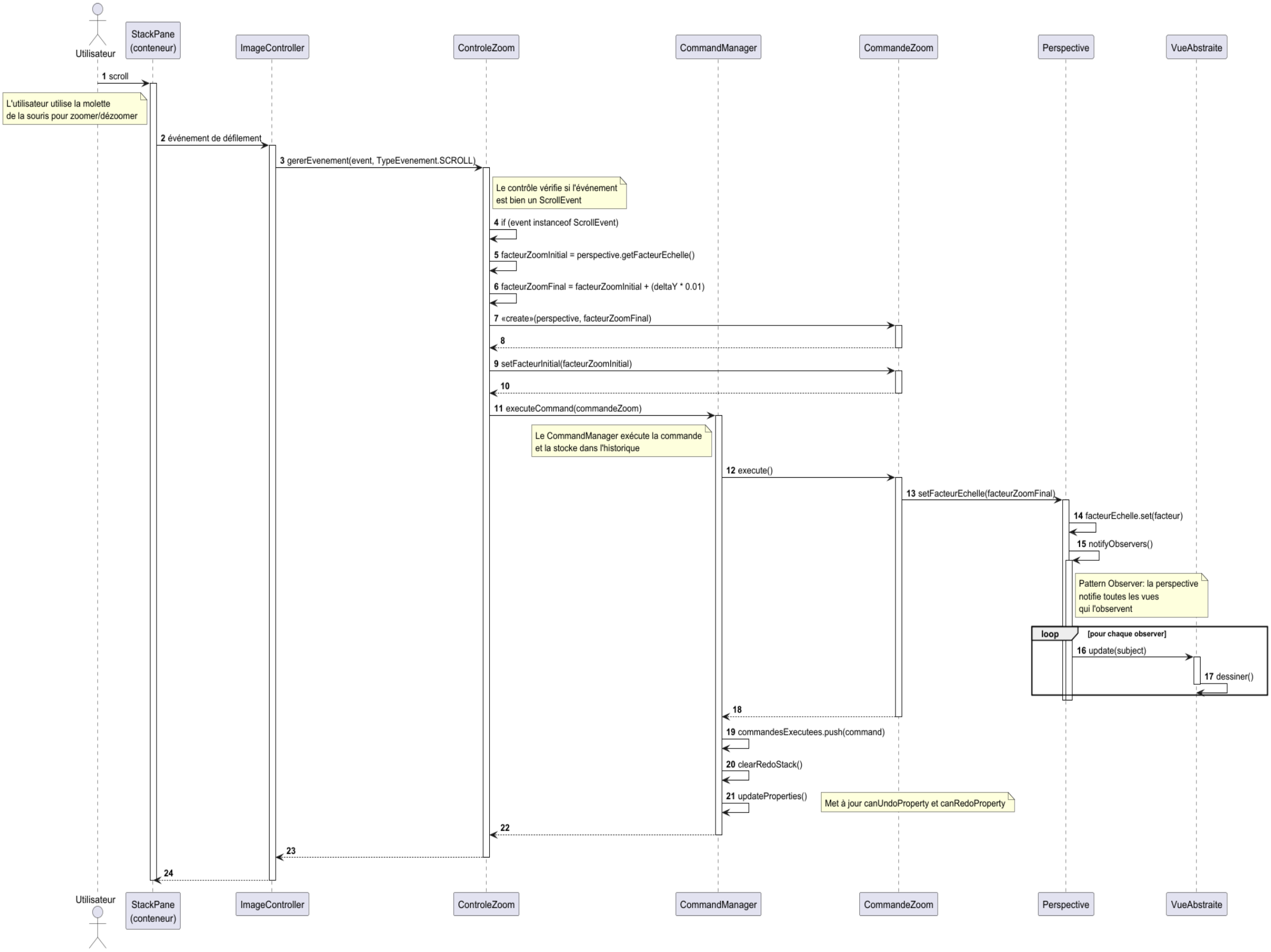

Diagramme de séquence de l'annulation d'une commande zoom

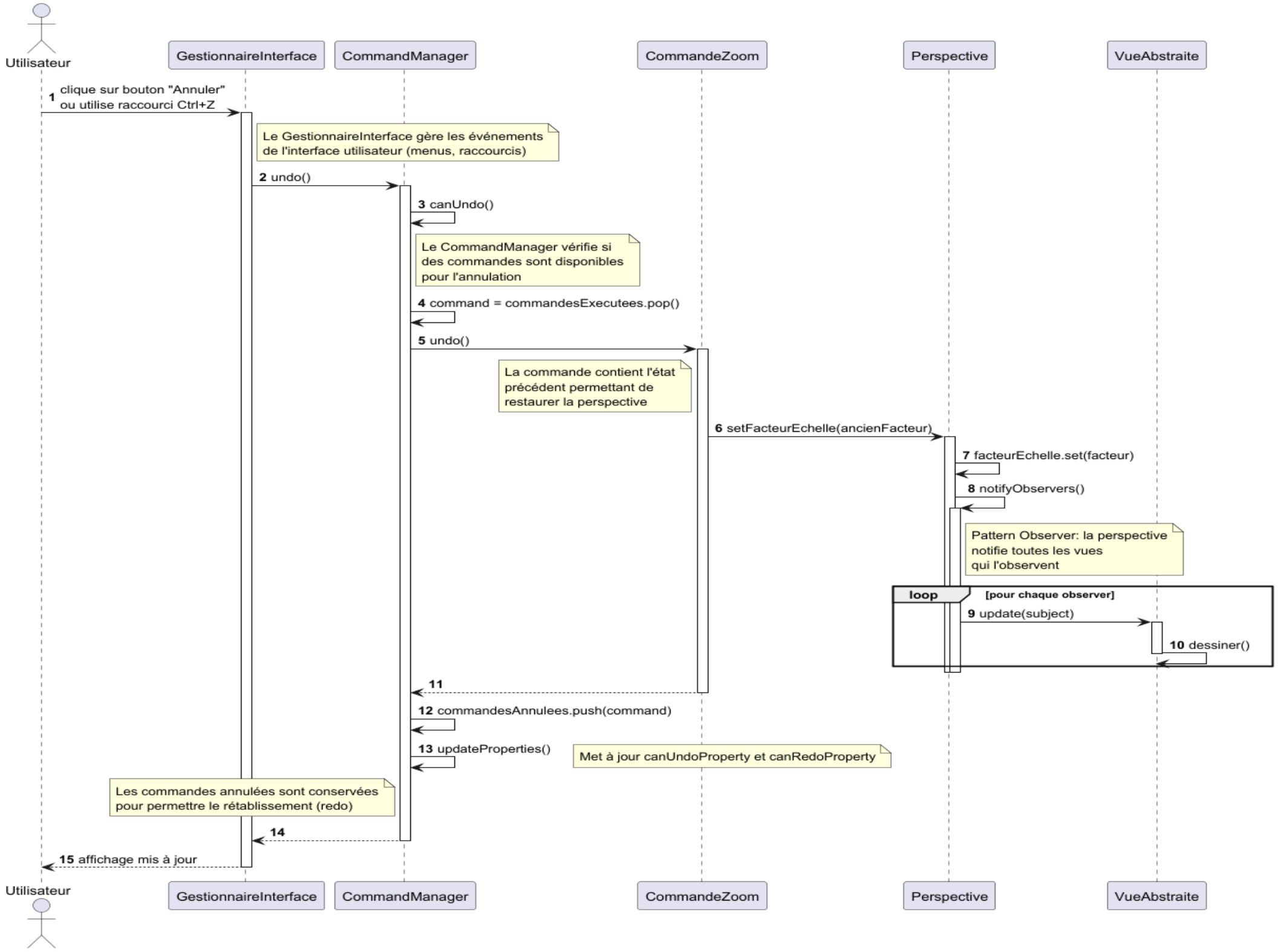

Tableau d'application des patrons de conception

Patron Observer

Élément du patron	Application dans le projet
Subject	`Perspective` et `Image` implémentent l'interface `Subject`
Observer	`VueAbstraite` et ses sous-classes implémentent l'interface `Observer`
Notify	Méthode `notifyObservers()` dans `Perspective` et `Image`
Update	`VueAbstraite.update(Subject)` qui appelle `dessiner()`
Attach	`perspective.attach(this)` dans le constructeur de `VueAbstraite`
Intérêt	Les vues se mettent à jour automatiquement quand la `Perspective` change

Patron Command

Élément du patron	Application dans le projet
Command	Interface `Command` avec méthodes `execute()` et `undo()`
ConcreteCommand	`ZoomCommand`, `TranslationCommand`
Invoker	`CommandManager` qui exécute les commandes
Client	`ControleZoom`, `ControleTranslation` qui créent les commandes
Receiver	`Perspective` dont l'état est modifié par les commandes
Intérêt	Permet l'historique d'actions avec undo/redo

Patron Strategy

Élément du patron	Application dans le projet
Strategy	Interface `ControleSouris`
ConcreteStrategy	`ControleZoom`, `ControleTranslation`
Context	`VueInteractive` qui utilise une stratégie
Client	`ImageController` qui configure les stratégies
Intérêt	Change dynamiquement le comportement des vues face aux événements souris

Patron Memento

Élément du patron	Application dans le projet
Memento	Classe `PerspectiveMemento`
Originator	`Perspective` qui crée et utilise des mementos
Caretaker	`Sauvegarde` qui stocke les mementos
Intérêt	Permet de sauvegarder et restaurer l'état des perspectives

Patron Singleton

Élément du patron	Application dans le projet
Instance	`CommandManager.instance`
GlobalAccessPoint	`CommandManager.getInstance()`
PrivateConstructor	`private CommandManager()`
Intérêt	Garantit une instance unique du gestionnaire de commandes

Patron MVC

Élément du patron	Application dans le projet
Model	`Image` et `Perspective`
View	`VueAbstraite` et ses sous-classes (`VuePrincipale`, `VueSecondaire`, `VueFixe`)
Controller	`ImageController`, `ControleZoom`, `ControleTranslation`
Intérêt	Sépare les données, leur présentation et les interactions utilisateur