Programme piloté par événement

M322 — Concevoir et implémenter des interfaces utilisateur

Jérôme Frosssard

EPAI

18 mai 2025

Événement

Un événement peut représenter :

- Une action de l'utilisateur (clic sur un bouton, caractère saisi dans une zone de texte, sélection dans une liste, etc.)
- Un événement du système (arrêt de l'ordinateur, alerte batterie faible, expiration d'un timer, etc.)

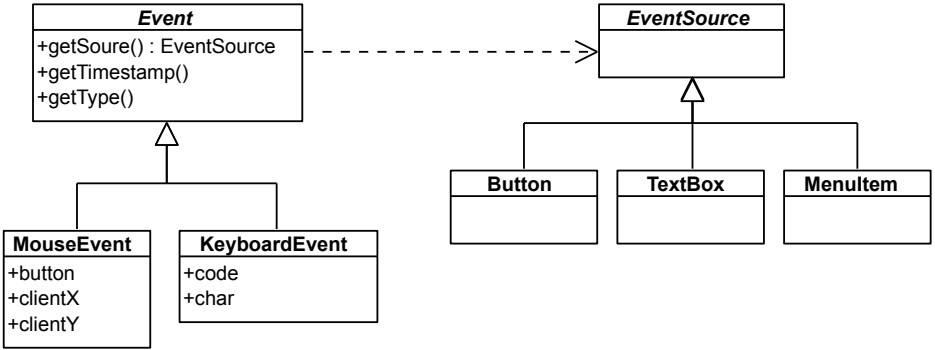
Un événement est décrit par :

- Un timestamp
- Son type (clic, sélection, survol du curseur, etc.)
- Sa source, c.-à-d. l'élément de l'interface (bouton, menu, etc.) sur lequel s'est portée l'action de l'utilisateur
- Des informations spécifiques à son type (position de la souris, état des boutons, etc.)

Représentation d'un événement

Dans un programme écrit avec un langage orienté objet (Java, JavaScript, etc.), les éléments de l'interface (widget) et les événements sont représentés par des objets.

Représentation des type d'objets (diagramme de classe) en UML :



Boucle d'événements

Un programme piloté par événement doit réagir lorsqu'un événement survient.

Les événements qui surviennent sont placés dans une file d'attente (*event queue*) par le processus chargé de la gestion des dispositifs d'entrée.

Le programme est construit autour d'une boucle appelée boucle d'événement (*event loop*) qui consiste à :

- Tant qu'il y a des événements, les traiter aussi rapidement que possible dans l'ordre d'arrivée.
- Attendre qu'un événement soit placé dans la file d'attente et recommencer.

Traitement des événements

Pour des raisons de fiabilité, les événements sont toujours traité l'un après l'autre, séquentiellement. Le même thread est utilisé pour la boucle d'événement, le traitement des événement et le rafraichissement de l'interface à l'écran.

Durant le traitement d'un événement, l'interface utilisateur ne pas être rafraîchie, le temps de traitement doit donc être aussi court que possible (quelques ms).

Pour assurer un traitement rapide, on peut utiliser :

- Des opérations d'E/S asynchrone et des promesses.
- Une bibliothèque permettant l'exécution de longs calculs à l'aide de worker threads (p. ex. Web Workers API).

Gestionnaire d'événements

Le traitement d'un événement est réalisé dans un sous-programme appelé gestionnaire d'événement (*event handler* ou *event listener*).

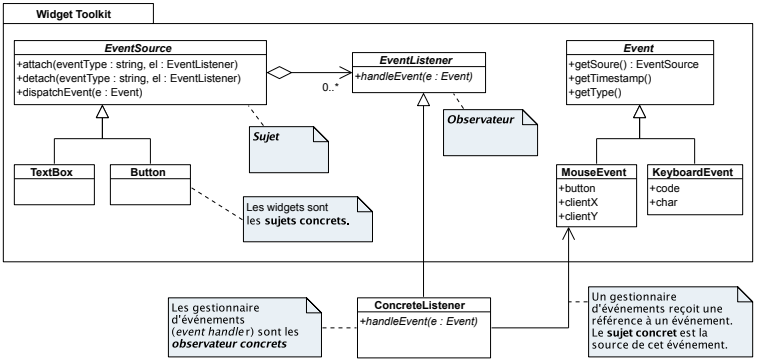
Les gestionnaires d'événements sont appelé dans le corps de la boucle d'événement. Un gestionnaire d'événements :

- Effectue le traitement des événements d'un ou plusieurs widgets appelé source d'événement (*event source*).
- Reçoit une description de l'événement en paramètre.

Distribution des événements

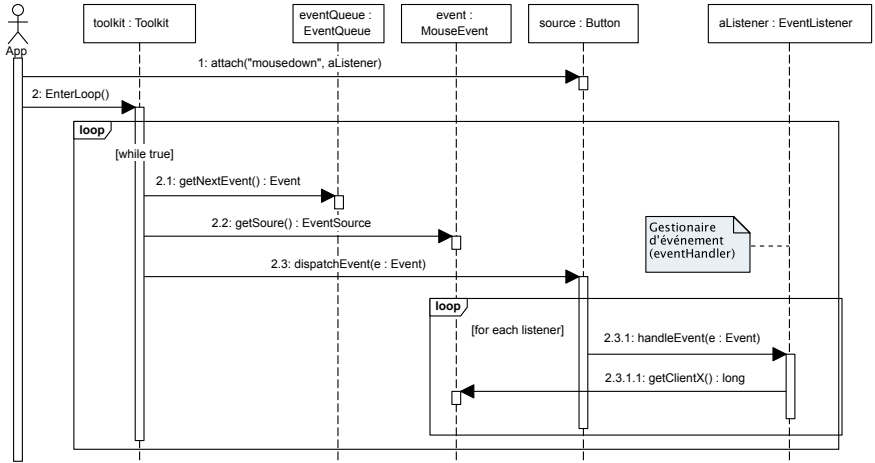
La distribution des événements (*event dispatching*) consiste à sélectionner le gestionnaire d'événement charger de traité un événement et à l'appeler.

La boucle d'événement et la distribution des événements sont généralement implémentées par le framework à l'aide d'une variation du patron observateur (*observer pattern*).

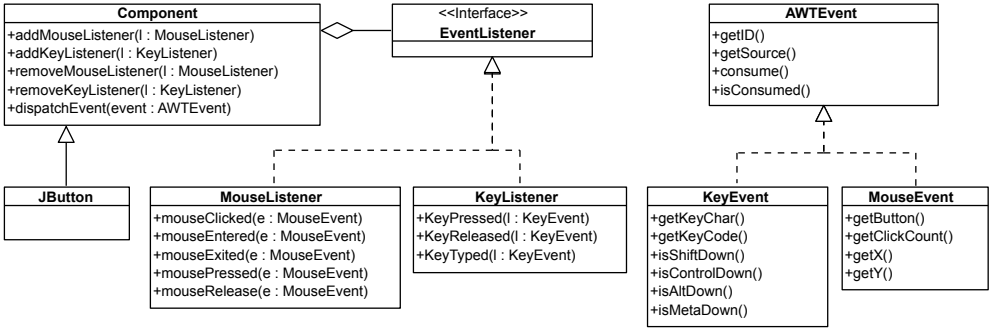


Boucle d'événement et distribution des événements

Diagramme de séquence simplifiée de la distribution des événements :



Vue simplifiée de l'implémentation du patron observateur dans le toolkit Java Swing :



Vue simplifiée de l'implémentation du patron observateur dans d'un navigateur Web avec JavaScript :