

Cours 2

Les événements, les filtres
d'événements et les gestionnaires
d'événements

Les événements

- Les événements sont générés lors d'une action de l'utilisateur sur l'interface.

Exple: clic sur un bouton, fermeture d'une fenêtre, sélection d'un élément de liste,....

- Un événement est sans conséquence si l'objet n'a pas de gestionnaire d'événements ou filtres d'événements enregistrés pour l'événement en question.

Qu'est qu'un événement JavaFX?

- C'est un objet de la classe `javafx.event.Event` ou une des sous-classes. Il se caractérise par les propriétés suivantes:
 - Une source
 - Une destination
 - Un type

Source d'un événement

- C'est l'origine de l'événement. Exple une fenêtre, une scène, un nœud. Noter que celle-ci change en fonction de sa position dans la chaine de répartition de l'événement (nous reviendrons plus tard sur le mécanisme de traitement d'un événement).

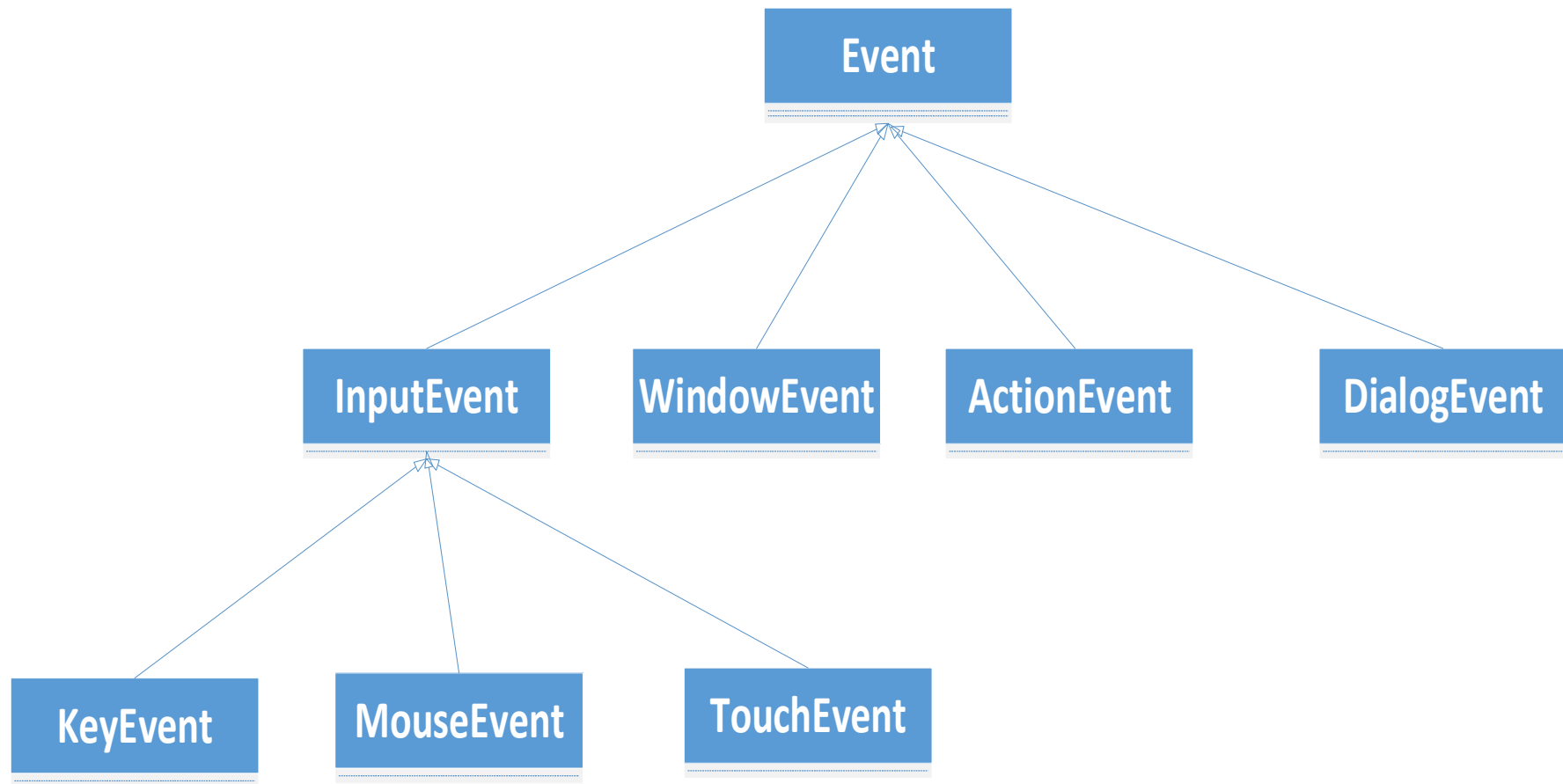
Destination d'un événement

- Elle détermine le chemin que l'événement doit parcourir pendant son traitement.
Exple, si un clic se produit sur un bouton, ce nœud sera la destination de l'événement `MOUSE_CLICKED`.

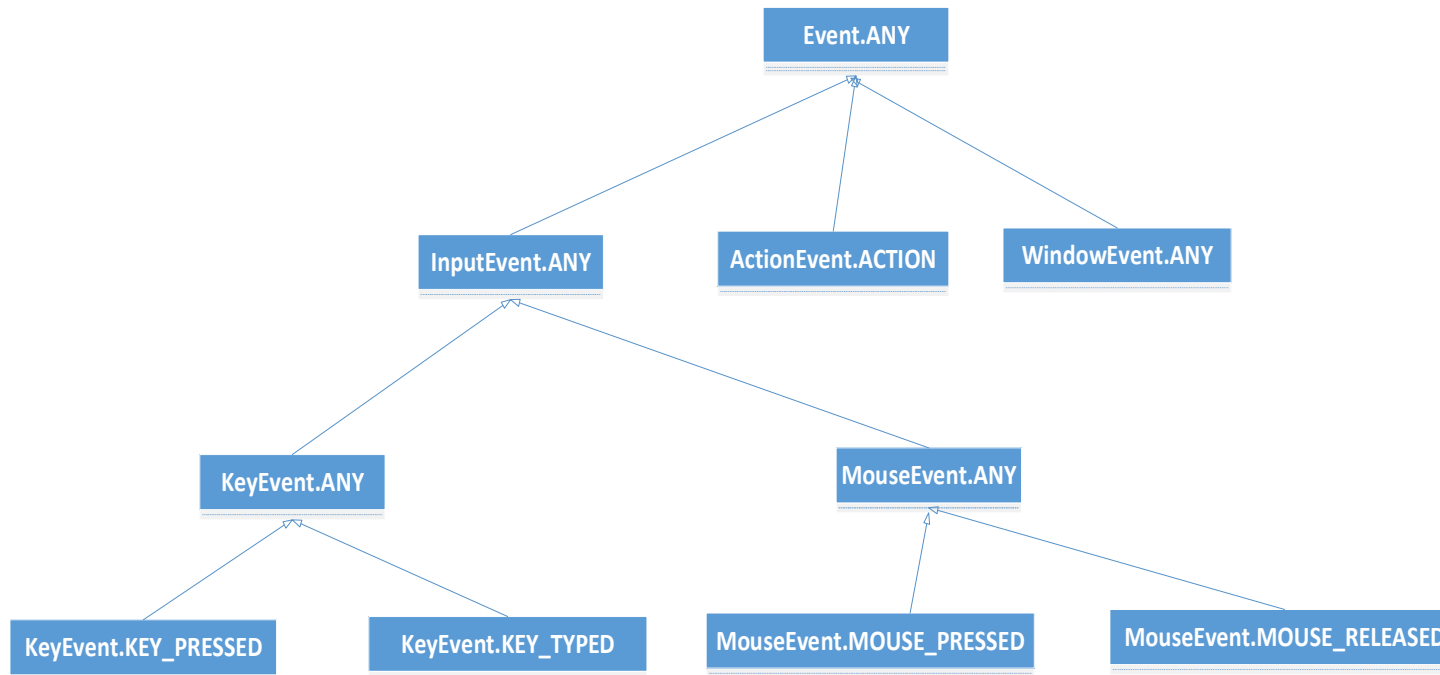
Type d'un événement

- Définit des types précis d'événements pour une classe donnée.
- Exemples
 - dans le cas de MouseEvent:
MOUSE_CLICKED, MOUSE_PRESSED,
MOUSE_RELEASED,.....
 - Dans le cas de la classe KeyEvent:
KEY_PRESSED, KEY_RELEASED,
KEY_TYPED,..

Une partie de la hiérarchie de la classe Event



Une partie de la hiérarchie de EventType: (pas par héritage de classes)



Traitement d'un événement

- Identifier le destinataire
- Définir le parcours
- Intercepter l'événement
- Faire remonter l'événement (Bubbling)

Identifier le destinataire

- Selon le type d'événement:
 - KeyEvent: nœud qui a le focus
 - MouseEvent: nœud à la position du curseur
 -

Définir le parcours

- Le parcours défini par défaut est:

fenêtre (Stage) ----Scene-----suite de
l'arborescence du graphe de scene jusqu'au
nœud où a eu lieu l'événement (exple clic de
souris)

Intercepter l'événement

- L'événement parcourt l'arborescence (racine---destination)
- Si un nœud du parcours a un filtre d'événements enregistré pour l'événement qui s'est produit, le filtre est exécuté et le parcours se poursuit.
- Si pas de filtres le parcours se poursuit normalement.
- L'événement arrive à la destination (où il sera traité) si aucun filtre n'a consommé l'événement.

Faire remonter l'événement

- L'événement remonte l'arborescence (destination----racine)
- Si un nœud du parcours a un gestionnaire d'événements enregistré pour l'événement qui s'est produit, le gestionnaire est exécuté et le parcours se poursuit.
- Si pas de gestionnaires, le parcours se poursuit normalement.
- L'événement arrive à la racine (Stage) si aucun gestionnaire n'a consommé l'événement et le traitement est terminé.

Gestionnaires d'événements/Filtres d'événements

- Objets de classes qui implémentent l'interface ***EventHandler<T extends Event>***
- Doivent être enregistrés pour l'événement auprès de nœuds.
- La différence réside dans le moment de leur exécution

Filtres d'événements

- **Filtre**: exécuté lors de l'interception d'un événement. Peut être utilisé pour un traitement commun à plusieurs nœuds enfants.
- Si plusieurs filtres sont enregistrés auprès d'un nœud, ceux-ci sont exécutés en allant du type d'événement le plus spécifique au plus générale. Exple: le filtre: ***MouseEvent.MOUSE_EVENT*** est exécuté avant ***InputEvent.ANY***
- Aucun ordre prédéfini pour les types d'événement de même niveau.

Gestionnaires d'événements

- **Gestionnaire**: exécuté lors de la remontée d'un événement. Peut être utilisé pour un traitement commun à plusieurs nœuds enfants.
- Si plusieurs gestionnaires sont enregistrés auprès d'un nœud, ceux-ci sont exécutés en allant du type d'événement le plus spécifique au plus générale.
Exple: le gestionnaire : ***MouseEvent.MOUSE_EVENT*** est exécuté avant ***MouseEvent.ANY***
- Aucun ordre prédéfini pour les types d'événement de même niveau. Cependant les gestionnaires enregistrés à l'aide de propriétés seront exécutés en dernier.

Interface EventHandler<T extends Event> extends EventListener

Voir le démonstrateur pour les
différentes façons de créer et d'enregistrer des gestionnaires/filtres
d'événements