

WIESLAW ZIELONKA  
ZIELONKA@IRIF.FR  
WWW.IRIF.FR/~ZIELONKA

# INTERFACES GRAPHIQUES

## key events

`KeyEvent` indique événement clavier. Les événements possibles :

- `KeyEvent.KEY_PRESSED`
- `KeyEvent.KEY_RELEASED`
- `KeyEvent.KEY_TYPED`
- `KeyEvent.ANY`

On peut intercepter les événement clavier en installant un handler sur un noeud soit par

```
node.addEventHandler(KeyEvent.KEY_TYPED, handler)
```

soit en utilisant les méthodes

- `onKeyPressed(handler)`
- `onKeyTyped(handler)`
- `onKeyReleased(handler)`

## focus

Le noeud recevra un `KeyEvent` s'il a un focus.

Mais certains composant, comme `Pane`, par défaut ne peuvent pas recevoir de focus. Pour qu'un `Pane` puisse recevoir `KeyEvents` il faut changer la propriété `focusTraversable`:

```
pane.setFocusTraversable(true);
```

Ensuite on peut demander le focus :

```
pane.requestFocus();
```

## key events — installer un handler

```
node.onKeyTyped( ( event ) -> {  
  
    String s = event.getCharacter();  
  
    //KeyEvent.getCharacter() retourne le  
    //caractère tapé sur le clavier, mais  
    // uniquement pour l'événement KEY_TYPED,  
    // CHAR_UNDEFINED pour KEY_PRESSED et  
    // KEY_RELEASED  
  
    // pour les touches de modification :  
    if(event.isControDown()){  
    if(event.isMetaDown()){  
    if(event.isAltDown()){  
    if(event.isShiftDown()){  
    if(event.isMetaDown()){
```



## KEY EVENTS - INSTALLER UN HANDLER

```
node.onKeyPressed( ( event ) -> {  
  
    KeyCode code = event.getKeyCode();  
  
    //getKeyCode() retourne le code d'une touche  
    // de clavier, voir la classe KeyCode pour  
    // les codes. C'est utile pour intercepter  
    // l'action sur les touches comme F11, LEFT  
    // etc, qui ne donnent pas de caractère  
    // affichable. C'est valable uniquement  
    // pour KEY_PRESSED et KEY_RELEASED.  
    // Pour KEY_TYPED la méthode retourne  
    // KeyCode.UNDEFINED
```