

ANNEXE 7 – Les Key Events

Nous avons surtout traité des événements souris jusqu'à présent. On peut également capter des objets `KeyEvents` à l'aide d'un écouteur mettant en œuvre l'interface `KeyListener`.

Les `KeyEvents` sont lancés lorsqu'un usager tape sur le clavier sur une source ayant le focus.

Il y a en fait 2 types de `KeyEvents` :

Ca tombe bien, l'interface `KeyListener` comprend 3 méthodes : `KeyPressed`, `keyReleased` et `keyTyped`

Comment faire pour identifier quel caractère ou quelle autre touche a été tapée ?

- ♣ pour un événement-caractère (`e.getID() == KeyEvent.KEY_TYPED`), on peut trouver le caractère avec : `getKeyChar()`

EX :

- ♣ pour un événement-touche de contrôle (`e.getID() == KeyEvent.KEY_PRESSED` ou `KeyEvent.KEY_RELEASED`), on peut trouver la touche tapée avec : `getKeyCode()`

EX :

EXERCICE : « Un début de Pac-Man »

Nous allons créer un GUI qui utilisera les notions de dessin vues avec les applets combinées avec des événements clavier.

- 1- Créez d'abord un JPanel (pas un JFrame) qui sera notre surface de dessin.
- 2- Un JPanel est une composante SWING, alors quelle méthode devez-vous redéfinir pour dessiner (voir exemple JApplet)
- 3- Dessiner un petit cercle, situé avec des coordonnées x et y (ce sera notre « pac-man »)
- 4- Codez des méthodes reculer, avancer, gauche, droite qui incrémenteront le x ou le y et redessineront le JPanel
- 5- Créez maintenant un JFrame, sur lequel vous ajouterez le JPanel (voir exercice sur les JApplets)
- 6- Faites la gestion d'événements clavier de manière à ce que :
 - a. un clic sur la lettre g entraîne un déplacement vers la gauche
 - b. un clic sur la lettre h entraîne un déplacement vers la droite
 - c. un clic sur la lettre y entraîne un déplacement vers le haut
 - d. un clic sur la lettre b entraîne un déplacement vers le bas
- 7- Vous pouvez refaire la gestion en utilisant cette fois-ci les flèches de direction