Interactions clavier et souris

Nous allons maintenant nous intéresser à l'interaction avec l'utilisateur. Pour l'instant (Processing offre énormément de possibilités) nous allons nous contenter de la souris et du clavier.

Souris

!!! info "Fonctions réservées" Processing propose 4 fonctions qui devront être complétées par le programmeur :

```
- le code se trouvant dans la fonction `mousePressed` est exécuté à chaque fois que l'utilis
- le code se trouvant dans la fonction `mouseReleased`  est exécuté à chaque fois qu'un bout
- le code se trouvant dans la fonction `mouseMoved`  est exécuté à chaque image tant que la
- le code se trouvant dans la fonction `mouseDragged`  est est exécuté à chaque image tant q

Ces 4 fonctions ne prennent aucun paramètre et ne retournent aucune valeur.

De plus les variables `mouseX` et `mouseY` permettent de récupérer la position de la souris.
```

!!! note "Exemple" "'python def setup(): global x, y, r size(400, 400) noStroke() fill(0) r = 10 x = 200 y = 200

```
def mousePressed():
    global x, y, r
    r = r + 100
    x = mouseX
    y = mouseY

def mouseReleased():
    global r
    r = r - 100

def draw():
    global x, y, r
    background(255)
    ellipse(x, y, 2*r, 2*r)

```

!!! example "{{ exercice() }}" Écrire un programme qui permettra d'afficher un nouveau carré à chaque clic de souris (sans effacer les carrés déjà présents). Le centre du carré devra se trouver au niveau du pointeur de la souris au moment du clic. La couleur du carré devra être aléatoire.

Clavier

!!! info “Fonctions réservées” Processing propose 2 fonctions qui devront être complétées par le programmeur :

```
- le code se trouvant dans la fonction `keyPressed` est exécuté à chaque fois que l'utilisat
- le code se trouvant dans la fonction `keyReleased` est exécuté quand une touche du clavier

Ces 2 fonctions ne prennent aucun paramètre et ne retournent aucune valeur.

De plus la variable `key` (ou `keyCode` pour les touches non ASCII[^1])  récupère la touche
```

!!! note “Exemple” “‘python def setup(): size(200, 200) textAlign(CENTER) background(255)

```
def keyPressed():
    if key == 'n':
        background(0)
        fill(255)
        text("NUIT", 100, 100)
    elif key == 'j':
        background(255)
        fill(0)
        text("JOUR", 100, 100)
    elif key == ENTER:
        background(255, 0, 0)
    elif key == ' ':
        background(0, 0, 255)
    elif key == CODED:
        if keyCode == UP:
            background(127)
            fill(0)
            text("Compris?", 100, 100)

def draw():
    pass
```

!!! example “{{ exercice() }}” Écrire un programme permettant d'afficher une balle noire qui pourra être déplacée à l'aide des flèches du clavier. La balle ne devra pas pouvoir sortir de la fenêtre.

!!! example “{{ exercice() }}” Reprendre l'animation de la balle qui rebondit sur les bords de la fenêtre et le modifier pour qu'un clic sur la balle change sa couleur et l'accélère.

```
Pour détecter si un clic a lieu sur la balle, il faut calculer la distance entre le centre d

On pourra utiliser la fonction suivante:
```

```python
def distance(x1, y1, x2, y2):
    return ((x1 - x2)**2 + (y1 - y2)**2) ** 0.5
```
Facultatif : on peut ajouter un compteur du nombre de clics sur la balle.