

## PARTIE 1: BOUTONS

## ! A RETENIR

Il y a **deux boutons A et B** sur la face avant de la carte Micro:bit.

On peut détecter quand ces boutons sont pressés, ce qui permet de déclencher des instructions sur l'appareil.

- `button_a.is_pressed()` : renvoie **True** si le bouton A est actuellement enfoncé et **False** sinon
- `button_a.was_pressed()` : renvoie **True ou False** pour indiquer si le bouton A a été appuyé depuis le démarrage de l'appareil ou la dernière fois que cette méthode a été appelée
- `button_a.get_presses()` : renvoie le nombre de fois où on a appuyé sur le bouton A

Avant de commencer, ouvrir le logiciel *Mu* et connecter votre carte Micro:bit sur un port USB.

## E EXERCICE 1 : PIXEL EN BALLADE

```

1 from microbit import *
2
3 x = 0
4 y = 0
5 while True:
6     display.set_pixel(x,y,0)
7     if button_a.was_pressed():
8         x = x-1
9     if button_b.was_pressed():
10        x = x + 1
11    x = max(0, min(x, 4))
12    display.set_pixel(x,y,9)
13    sleep(20)

```

1. Flasher ce programme sur la carte puis expliquer ce qu'il fait.
2. Quel est le rôle de la commande `x = max(0, min(x, 4))` en ligne 11 ?
3. Modifier ce programme pour que le pixel se balade cette fois-ci sur la première colonne.

## E EXERCICE 2 : CLIGNOTANTS

```

1 from microbit import *
2
3 while True:
4     if button_a.is_pressed():
5         ...
6     if button_b.is_pressed():
7         ...

```

1. Compléter ce programme afin qu'il affiche une flèche clignotante vers la gauche lorsque le bouton A est enfoncé et une flèche clignotante vers la droite lorsque le bouton B est enfoncé.
2. Flasher le programme sur la carte et vérifier qu'il fonctionne.
3. Modifier ce programme pour afficher en plus une flèche de recul (vers le bas) lorsque les deux boutons sont enfoncés.  
Indication : On pourra modifier les premières instructions conditionnelles pour indiquer qu'un seul bouton est enfoncé.

APPEL

→ Appeler le professeur pour vérification

## ? CAPTEUR DE LUMIÈRE

En inversant les LEDs d'un écran pour devenir un point d'entrée, l'écran **LED** devient un capteur de lumière basique, permettant de détecter la luminosité ambiante.

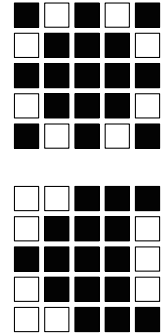
La commande `display.read_light_level()` retourne un entier compris entre 0 et 255 représentant le niveau de lumière.

## E EXERCICE 3 : LE SOLEIL A RENDEZ-VOUS AVEC LA LUNE

```

1 from microbit import *
2
3 soleil = Image("90909:"
4               "09990:"
5               "99999:"
6               "09990:"
7               "90909")
8
9 lune = Image("00999:"
10             "09990:"
11             "09990:"
12             "09990:"
13             "00999")
14
15 while True:
16     if display.read_light_level()>...:
17         display.show(soleil)
18     else:
19         display.show(...)
20         sleep(10)

```



1. Compléter le programme ci-dessus pour afficher une image de lune si on baisse la luminosité (en recouvrant la carte avec sa main par exemple) et un soleil sinon.
2. Modifier le programme pour qu'il affiche en défilement le niveau de luminosité ambiant après le symbole obtenu (soleil/lune).

## ? CAPTEUR DE TEMPÉRATURE

La carte Micro:bit n'a pas un capteur de température dédié. Au lieu de cela, la température fournie est en fait la température de la **puce de silicium du processeur principal**. Comme le processeur chauffe peu en fonctionnement (c'est un processeur ARM à grande efficacité), sa température est une bonne **approximation** de la température ambiante.

L'instruction `temperature()` renvoie la température de la carte Micro:bit en degrés Celsius.

## E EXERCICE 4 : THERMOMÈTRE DE SECOURS

Écrire un programme qui affiche la température en défilement.

**APPEL**

→ Appeler le professeur pour vérification



## CAPTEUR DE MOUVEMENT

Un **accéléromètre** mesure l'accélération de la carte Micro:bit, ce composant détecte quand la carte est en mouvement. Il peut aussi détecter d'autres actions (gestes), par exemple quand elle est secouée, inclinée ou qu'elle tombe.

Des accéléromètres sont présents dans les smartphones et les manettes de jeux afin de pouvoir afficher l'image dans le bon sens, ou bien de changer de direction dans un jeu.

L'accéléromètre mesure le mouvement selon trois axes :

- X - l'inclinaison de gauche à droite
- Y - l'inclinaison d'avant en arrière
- Z - le mouvement haut et bas

L'instruction `accelerometer.get_x()` permet de détecter un mouvement de gauche à droite en renvoyant un nombre compris entre -1023 et 1023 (0 étant la position "d'équilibre").

L'instruction `accelerometer.is_gesture('shake')` renvoie **True** ou **False** selon si la carte est secouée.



### EXERCICE 5 : EN HAUT, EN BAS, À GAUCHE, À DROITE

```

1 from microbit import *
2
3 while True:
4     abscisse = accelerometer.get_x()
5     if abscisse > 500:
6         display.show(Image.ARROW_E)
7     elif abscisse < -500:
8         display.show(Image.ARROW_W)
9     else:
10        display.show("-")

```

1. Flasher ce programme sur la carte. Expliquer ce qu'il fait.
2. Compléter le programme pour qu'il affiche une flèche vers le haut si on incline la carte en avant et une flèche vers le bas si on l'incline en arrière.



### EXERCICE 6 : LANCER DE DÉS

Écrire un programme qui simule un dé en affichant une face au hasard lorsque la carte Micro:bit est secouée.

**APPEL**

→ Appeler le professeur pour vérification



## CAPTEUR DE DIRECTION

La **boussole** détecte le champ magnétique de la Terre, nous permettant de savoir quelle direction la carte Micro:bit indique.

La boussole doit être étalonnée avant de pouvoir être utilisée. Pour cela, on utilise `compass.calibrate()` qui exécute un petit jeu. Au départ, la carte fait défiler "Tilt to fill screen" puis il faut incliner la carte pour déplacer le point au centre de l'écran autour jusqu'à remplir la totalité de l'écran.

La fonction `compass.heading()` donne le cap de la boussole sous la forme d'un entier compris entre 0 et 360, représentant l'angle en degrés, dans le sens des aiguilles d'une montre, avec le nord égal à 0.

## E EXERCICE 7 : ROSE DES VENTS

```
1 from microbit import *
2
3 compass.calibrate()
4
5 while True:
6     if compass.heading() < ... or compass.heading() > ...:
7         display.show(Image.ARROW_N)
8     else:
9         display.show(Image.DIAMOND_SMALL)
```

1. Compléter le programme ci-dessus afin qu'il indique le Nord.
2. Améliorer le programme pour que la carte Micro:bit indique "N", "S", "E" et "O" en fonction de l'orientation de la boussole.

**APPEL**

→ Appeler le professeur pour vérification

## P POUR ALLER PLUS LOIN

### Jeu 1 : Magic8Ball.py

- Copier le programme suivant appelé Magic8Ball.py sur la carte Micro:bit.
- Modifier son code pour fabriquer un dé électronique à six faces .

```
from microbit import *
import random
reponses = ["OUI", "NON", "SUREMENT", "PAS SUR"]
while True:
    display.show("8")
    if accelerometer.was_gesture("shake"):
        display.clear()
        sleep(1000)
        display.scroll(random.choice(reponses), delay=80)
```

### Jeu 2 : SimpleSlalom.py

- Copier le programme suivant appelé SimpleSlalom.py sur la carte Micro:bit.
- Modifier son code pour fabriquer un dé électronique à six faces .

```
from microbit import *
import random
reponses = ["OUI", "NON", "SUREMENT", "PAS SUR"]
while True:
    display.show("8")
    if accelerometer.was_gesture("shake"):
        display.clear()
        sleep(1000)
        display.scroll(random.choice(reponses), delay=80)
```