Activité Microbit n°2

Informatique embarquée

Seconde SNT Lycée du Parc

1 Manipulation d'images

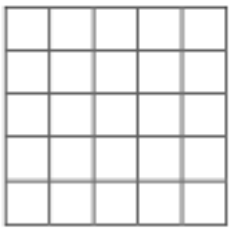
Exercice 1

Créez votre propre image !

Chacun des 25 pixels du micro:bit peut prendre une valeur entre 0 et 9 : 0 pour éteint, 9 pour allumé avec la puissance maximale, les valeurs comprises entre 1 et 8 sont des valeurs de luminosité intermédiaires. Par exemple, saisir le programme ci-dessous dans l'éditeur Mu Python puis le transférer sur la carte micro:bit.
Devinez ce que peut représenter l'image affichée.

```
from microbit import *
boat = Image("05050:"
"05050:"
"05050:"
"99999:"
"09990")
display.show(boat)
```

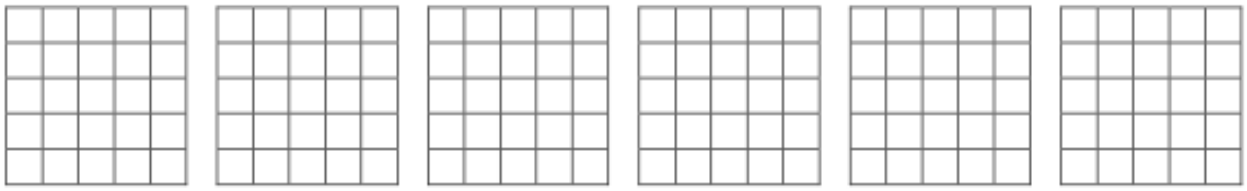
Dessinez à l'aide de la grille ci-dessous votre propre image et testez-là !



Exercice 2

Créez votre propre animation

1. Pour cela, comme un dessin animé, créez une liste d'images qui vont s'afficher successivement. Il faut en créer au moins 2.



Par exemple, si vous avez créé 6 bateaux nommés `boat1, boat2, boat3, boat4, boat5, boat6`, vous pouvez alors afficher l'animation à l'aide des instructions suivantes. La liste ordonnée d'images est créée et les images défilent toutes les 200 millisecondes.

```
all_boats = [boat1, boat2, boat3, boat4, boat5, boat6]
display.show(all_boats, delay=200)
```

2. Modifiez le programme pour qu'il affiche l'animation quand on appuie sur le bouton A et l'animation « inverse » quand on appuie sur le bouton B.

2 Simuler le hasard

Exercice 3

1. La fonction `randint(a, b)` du module `random` de `Python` permet de simuler le choix d'un entier aléatoire compris entre deux entiers `a <= b`. On commence par l'importer avec l'instruction `from random import randint`.

Compléter le programme ci-dessous dans l'éditeur Mu Python puis le transférer sur la carte micro:bit pour que la simulation d'un lancer de dé à 6 faces soit affichée pendant deux secondes sur l'écran dès qu'on appuie sur le bouton A.

```
from microbit import *
from random import randint

while True:
    #à  compléter
    display.clear()
```

2. Adapter le programme précédent pour que le résultat de chaque lancer soit affiché jusqu'à l'obtention du premier 6 puis affiche l'image `HAPPY` pendant deux secondes avant que la partie se termine.

```
from microbit import *
from random import randint

#à  compléter
display.show(Image.HAPPY)
sleep(2000)
```

3. Modifier le dernier programme pour que le nombre de lancers avant l'obtention du premier 6 soit affiché à la fin pendant deux secondes (après l'image `HAPPY`).

Exercice 4

1. Saisir le programme ci-dessous dans l'éditeur Mu Python puis le transférer sur la carte micro:bit. On note A un appui sur le bouton A, B un appui sur le bouton B et AB un appui simultané sur A et B.
   - Décrire la succession d'affichages obtenus pour la séquence d'appuis A - A - B - A - AB
   - Décrire la succession d'affichages obtenus pour la séquence d'appuis A - B - B - B - AB
   - Combien de séquences de six appuis (sans compter l'appui simultané AB) permettent d'obtenir l'affichage -2.

```
from microbit import *
from random import randint

validation = False
choix = 0
while not validation:
    boutonA = button_a.was_pressed()
    boutonB = button_b.was_pressed()
    if boutonA and boutonB:
        validation = True
    elif boutonA:
        choix = choix + 1
    elif boutonB:
        choix = choix - 1
    display.show(choix)
    sleep(1000)
display.show(Image.HAPPY)
```

La variable `validation` est de type *booléen*, elle peut prendre deux valeurs `True` ou `False`. Les valeurs *booléennes* sont utilisées dans les opérations logiques comme la **conjonction** (ET) et la **disjonction** (OU).

Exercice 5

À l'aide des programmes précédents, écrire un programme qui simule le jeu de devinette décrit ci-dessous :

- l'ordinateur choisit un nombre au hasard entre 1 et 9 ;
- tant que le joueur n'a pas deviné ce nombre, il peut proposer un nombre à l'aides boutons A et B : un appui sur A permet d'incrémenter de 1 la proposition précédente (0 avant que la partie commence) et un appui sur B de la décrémenter de 1, un double appui sur A et B valide la proposition ;
- le jeu s'arrête lorsque le joueur a deviné le nombre secret, l'image `HAPPY` est affichée ainsi que le nombre de coups pour deviner le nombre secret.

Quel nombre minimal de coups nous permet de deviner le secret à coup sûr ? Comparer cette méthode avec une recherche ordonnée si on doit deviner un nombre entre 0 et 100000.

XKCD 1613 : The Three Laws of Robotics