APMEP Toulouse IRES Numérique Les Maths en Scène

La carte Micro:bit et Python

Nous utiliserons, dans les activités proposées, l'interface de Vittascience:
https://fr.vittascience.com/microbit/?link=5d5c10ff185bf&console=bottom&mode=code
Cette interface permet notamment de faire fonctionner les programmes sur une carte micro:bit virtuelle.

Instructions de base

- from microbit import* permet d'importer la librairie liée à la carte microbit.
- while True permet de faire des boucles qui tournent à l'infini
- sleep(1000) permet de faire une pause de 1000 millisecondes dans le programme.

Partie 1: Découverte de la matrice LED

*Activité 1

Recopier et tester le programme ci-dessous (pour le tester il faut appuyer sur le triangle noir en haut à droite de l'écran: "Démarrer le simulateur")

```
1 from microbit import*
2 display.show("Bonjour")
```

- Que permet de faire l'instruction display.show("Bonjour") ?
- Tester l'instruction suivante: display.scroll("Bonjour"). Quelle différence y a-t-il avec display.show() ?

Pour que le message tourne en boucle, on peut rajouter: display.show("Bonjour", loop=True) *(Ne pas hésiter à tester).*

*Activité 2

Que fait le programme suivant ?

```
1 from microbit import*
2 while True:
3   display.show("A")
4   sleep(500)
5   display.show(" ")
6   sleep(500)
```

A faire: Modifier le programme précédent pour qu'il compte en boucle de 0 à 4.

*Activité 3 Images et matrice LED.

La matrice LED de la carte Micro:bit permet d'afficher des images. De nombreuses images sont déjà prédéfinies dans Micropython.
Vous trouverez une liste ici: https://microbit-micropython.readthedocs.io/fr/latest/tutorials/images.html

A faire: Tester par exemple display.show(Image.HEART) puis faire afficher un smiley qui sourit.

Partie 2: Boutons et matrice LED

Utilisation des boutons

Il y a deux boutons intégrés sur la carte micro:bit accessibles via les objets button_a et button_b. Ces objets permettent l'écriture de programmes nécessitant une interaction avec l'utilisateur.

On peut utiliser:

- button_a.is_pressed() qui renvoit True si le bouton A est pressé au moment où la méthode est invoquée.
- button_a.was_pressed() permet de savoir si le bouton A a été pressé pendant que le programme effectuait une autre tâche.

*Activité 4 : Borne de satisfaction

Première étape: Recopier et compléter le programme ci-dessous pour qu'il affiche:

- un smiley qui sourit (Image.HAPPY) si le bouton A est pressé
- un smiley triste (Image.SAD) si le bouton B est pressé

```
1 from microbit import*
2
3 while True:    #boucle infinie
4   if button_a.was_pressed():
```

Deuxième étape: Compléter votre programme précédent pour qu'il affiche:

- un message d'accueil qui pose la question "Etes-vous satisfait ?"
- le symbole choisi en fonction du bouton utilisé
- une image (YES) ou un message du type "Merci" lorsque la personne a voté.

Votre programme devra également faire les instructions en boucle pour que l'on puisse voter les uns après les autres.

Aide: Rajouter des temps d'attente entre les actions en utilisant sleep (temps d'attente en millisecondes)

Pour aller plus loin : Fabriquer une "vraie" borne de satisfaction.
La commission inter IREM TICE propose de prolonger cette activité en faisant des branchements sur les pins d'entrée/sortie d'une carte micro:bit dans le but de fabriquer un vrai boitier de vote: http://tice.univ-irem.fr/?p=4161
On pourra également rajouter un compteur permettant de comptabiliser les différents votes.

Partie 3: Capteur de température

La carte Micro:bit dispose d'un capteur de température.

Tester le programme suivant qui permet d'afficher la température en degré. *Sous la carte micro:bit virtuelle, vous pouvez modifier la température.*

```
1 from microbit import*
2 while True:
3   temp= temperature()
4   display.scroll(str(temp)+'C')
```

Remarque: temp et 'C' n'étant pas du même type, il faut ici convertir le nombre contenu dans la variable temp en chaîne de caractère. On utilise pour cela l'instruction str.

*Activité 5: Affichage de la température en degrés Celsius ou en degrés Fahrenheit.

Compléter le programme précédent pour qu'il affiche:

- la température en degrés Celsius si on appuie sur le bouton A.
- la température en degrés Fahrenheit si on appuie sur le bouton B : F = 1.8*temp+32

*Activité 6: Alerte Canicule

En Europe de l'Ouest, il est considéré qu'une canicule correspond à une température de nuit supérieure à 18-20 °C et une température de jour supérieure à 30-35 °C. On choisira ici un seuil de 30 degrés Celsius.

Le but est d'écrire un programme qui permet d'alerter si une température seuil est dépassée.

Première étape

Créer une variable seuil qui prendra la valeur 30.
Si la température dépasse la valeur seuil, afficher une croix (Image.NO) sinon afficher un check (Image.YES)
(Pour tester votre programme, faites varier la température dans la console)

Deuxième étape

On souhaite pouvoir ajuster la valeur du seuil en utilisant les boutons A et B.
Compléter votre programme:
Si on appuie sur le bouton A, le seuil augmente de 1 degré et le nouveau seuil est affiché.
Si on appuie sur le bouton B, le seuil diminue de 1 degré et le nouveau seuil est affiché.

Liens vers des ressources exploitables en classe (liste non exhaustive)

La version française du site de micro:bit: https://microbit-micropython.readthedocs.io/fr/latest/tutorials/introduction.html
En anglais (plus complet): https://microbit.org/projects/
Le site MyScenari où l'on peut trouver de très nombreuses idées de projets: https://lecluseo.scenari-community.org/CircuitPython/co/module_Micropython_3.html
Le site de la commission inter IREM TICE: http://tice.univ-irem.fr/?page_id=906
Le site de la DANE de Besançon: https://pedagogie-numerique.ac-besancon.fr/2019/06/differentes-facons-de-programmer-en-python-avec-une-microbit/#9b
Ressources de Vittascience: https://fr.vittascience.com/learn/?search=&support=1&difficulty=0,1,2,3&lang=fr,en
Autre interface permettant d'utiliser une carte micro:bit virtuelle: create.withcode.uk