



Le jeu du tir à la corde : Micro:Bit

1. Introduction

Dans cet exemple d'application, l'objectif est, selon le principe du jeu du tir à la corde, tirer un pixel vers soi en pressant un bouton avec une fréquence de frappe la plus importante. Le joueur ayant la fréquence la plus importante déplace le pixel vers lui. Le rafraîchissement se fera toutes les 5 secondes.



Nous allons utiliser le logiciel MU Editor.

MU editor est un logiciel permettant de déposer directement le microprogramme sur la carte, sans avoir à passer par l'étape manuelle de dépôt du fichier .HEX et il permet également de recevoir et d'envoyer des données en temps réel à la carte (on appelle cela la **console REPL**). Aller sur le site <https://codewith.mu/> et suivre les instructions pour l'installation. Ouvrir **MU editor**.

(Pour les versions inférieure à windows10, il faut également installer le Windows Serial driver :

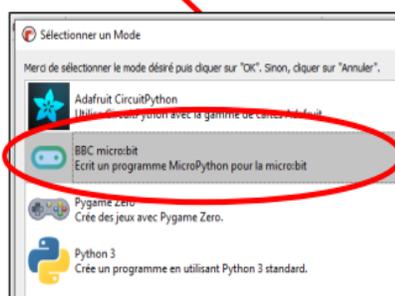
<https://os.mbed.com/docs/mbed-os/v5.7/tutorials/windows-serial-driver.html>)



Dépose le microprogramme sur la carte.

Ouvre la console pour afficher les mesures (console)

Permet de vérifier les erreurs de codage (debuggage).



- 1) Choisir "mode" puis "BBC micro:bit".
- 2) Faire "nouveau", puis "enregistrer" pour sauvegarder.
- 3) Taper le code.
- 4) Faire "vérifier" et suivre les conseils données en cas d'erreurs ou de problèmes de mises en forme du code.
- 5) Déposer le microprogramme sur la carte : "flasher" Le programme démarre, faire "REPL" pour afficher la console si besoin. Dans ce cas il faut appuyer sur le bouton RESET de la carte pour relancer le programme et l'affichage.

Remarque : Pour chaque nouvelle modification du programme, il faut fermer la console « REPL », et « flasher » de nouveau pour déposer le programme modifié sur la carte.



2. L'affichage sur l'écran

- Toujours commencer un programme par la ligne *from microbit import **
- Ecrire le programme suivant (**Attention à respecter les minuscules/majuscules et les espaces**)

La carte microbit est composée de plusieurs LED (5*5)qui permettent de créer un petit écran.

2.1 Affichage statique.

```
from microbit import *
```

```
display.show('B')
```



- *Flasher le programme dans la carte et choisir la microbit*

Afficher une chaîne de caractère.

La syntaxe est la suivante :

```
display.show( 'x' , delay = 400, wait= False, loop = False, clear = False)
```

- ✚ x correspond à l'information à afficher
- ✚ delay correspond au temps d'affichage fixe en millisecondes de chaque éléments.
- ✚ Loop : si True l'affichage se fait en boucle.
- ✚ Clear : si True l'affichage est effacée à la fin.

Créer un programme qui permet d'afficher en boucle 'Bienvenue en SNT' qui dure 16 secondes.

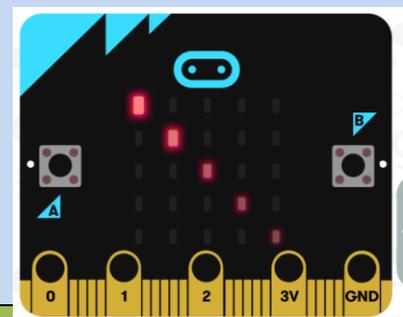
2.1 Allumer une LED

Il est possible de commander le flux lumineux des différentes LED de l'écran avec la commande suivante :

```
display.set_pixel ( x , y , intensité)
```

- ✚ x est la position sur la ligne :
- ✚ y est la position dans la colonne :
- ✚ intensité : 0 éteint , 9 intensité max.

Ecrire le programme permettant d'avoir le résultat suivant :



2.2 Faire clignoter l'écran

Commandes pour agir sur l'écran :

- ✚ Pour effacer l'écran : `display.clear ()`
- ✚ Pour allumer l'écran : `display.on()`

Créer le programme suivant en vous aidant des liens hypertexte :

Début

Tant que vrai

Afficher l'image suivante

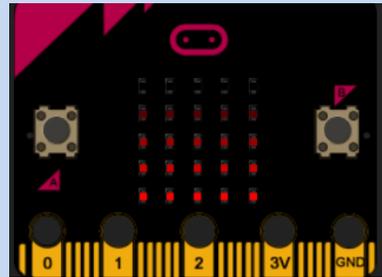
Attendre 200ms

Eteindre l'afficheur

Attendre 200ms

Fin tant que

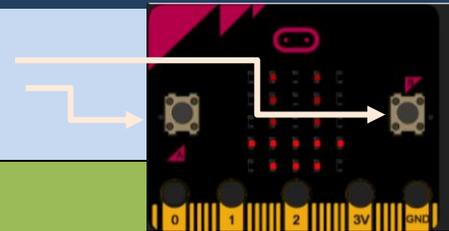
Fin



```
from microbit import *
```

3. Les boutons

La carte micro-bit possède deux boutons poussoirs sur son écran.



```
from microbit import *
```

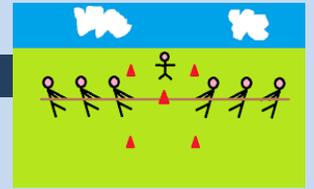
```
sleep(10000)
```

```
display.scroll(str(button_a.get_presses()))
```

Que fait ce programme ?



4. Le jeu du tir à la corde



Pendant la partie, chaque joueur peut presser quand il veut son boutons poussoirs.

Le gagnant est celui qui réussira à attirer le pixel de son côté de l'écran.

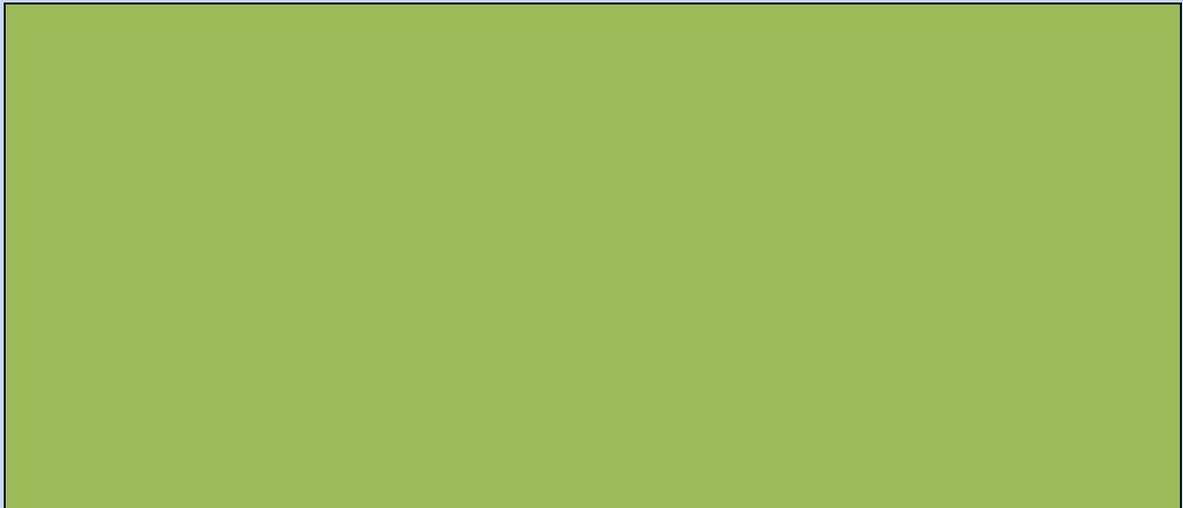
L'écran affichera le vainqueur à l'aide d'une variable « gagnant », ainsi que le nombre de pressions total sur le bouton effectué au cours de la partie.

4.1 Ecrire un programme qui permet de compter combien de fois on a appuyé sur le bouton a et b pendant 5 secondes. Afficher le résultat.



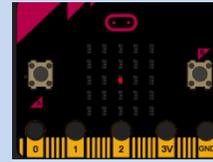
4.2 Même programme en ajoutant le nom du vainqueur.

Utiliser des instructions conditionnelles ([if, elif ou else](#)).

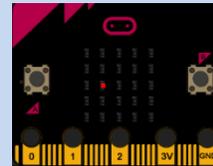


4.3 Programme final

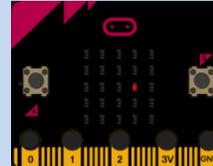
On va réaliser une animation visuelle.



Voici la position du pixel avant de démarrer la partie.



Après 5 secondes, l'équipe A à gagner une manche.



Si c'est l'équipe B le pixel se déplace à droite.

La partie est gagnée si une équipe gagne 3 manches consécutives.

Voici l'algorithme du programme du jeu du tir à la corde.

DÉBUT

Gagnant ← « X »

TANT QUE *gagnant* = « X »

Attendre 1 s

Nb_pressions_boutonA ← nombre de pressions sur le bouton A

Nb_pressions_boutonB ← nombre de pressions sur le bouton B

SI *nb_pressions_boutonA* > *nb_pressions_boutonB* *ALORS*

SI *position* < 1 *ALORS*

Gagnant ← « A »

SINON

Position ← *position* - 1

FIN SI

SINON SI *nb_pressions_boutonA* < *nb_pressions_boutonB* *ALORS*

SI *position* > 4 *ALORS*

Gagnant ← « B »

SINON

Position ← *position* + 1

FIN SI

FIN SI

Afficher pixel en position

FIN TANT QUE

Afficher gagnant

FIN



Compléter le programme du jeu du tir à la corde.

```
from microbit import *  
gagnant = 'X'  
position = 2  
display.set_pixel (position, 2 , 9)  
while gagnant == 'X':  
    sleep( 5000)
```

