

Dans cette activité vous allez découvrir la carte microbit :

<https://www.youtube.com/watch?v=4I31LLKwPtk>


C'est une carte programmable pas cher (16€), qui dispose d'un émulateur en ligne :

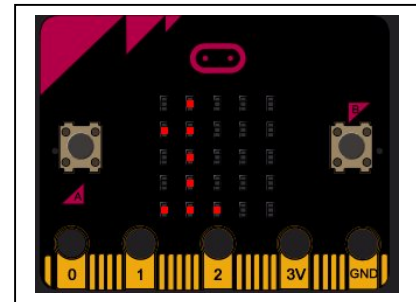
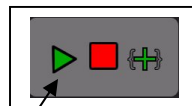
(Pour le lien → me contacter <http://www.ile-reunion.org/louispayen/snt/> )


Toute la classe va donc pouvoir la programmer même s'il n'y pas suffisamment de carte pour tout le monde ☺

La programmation se fait en python. Voyons tout de suite des petits programmes d'exemples :  
Allez sur ce lien : (Pour le lien → me contacter ) puis taper le code ci-dessous :


### Programme 1 :

```
mycode.py 
1 from microbit import *
2 while True:
3     display.show("1")
4     sleep(500)
5     display.show(" ")
6     sleep(500)
7
```




Lancer le programme en cliquant sur  et vous verrez la carte Microbit apparaître et le programme s'exécuter (affichage d'un « 1 » pendant 500ms puis effacement pendant 500ms , ce qui donne un effet clignotant )

### Programme 2 :

```
mycode.py 
1 from microbit import *
2 while True:
3     for i in range (9):
4         display.show(str(i))
5         sleep(500)
6         display.show(" ")
7         sleep(500)
8
```

Le programme permet de faire défiler les chiffres de 0 à 8.  
Modifier la ligne 3 pour qu'il défile jusqu'à 9  
On notera la présence de str(i) qui permet de transformer le nombre i en une chaîne de caractères


### Programme 3 :

```
mycode.py 
1 from microbit import *
2 while True:
3     for i in range (9,0,-1):
4         display.show(str(i))
5         sleep(500)
6         display.show(" ")
7         sleep(500)
8
```

Que fait ce programme ?

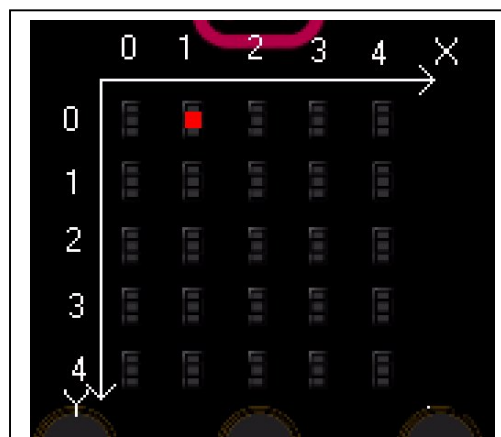
Pouvez vous modifier la ligne 3 pour que le décomptage se fasse correctement jusqu'à 0 ?

### Programme 4 :

```
mycode.py 
1 from microbit import *
2 while True:
3
4     display.set_pixel(1,0,9)
5
```

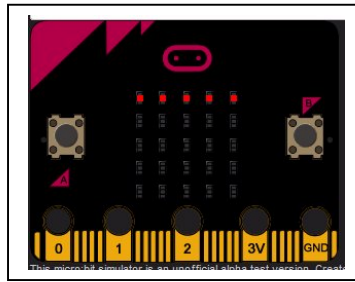
X,Y, Luminosité

Luminosité : 0 à 9 (max)



**Programme 5 :**

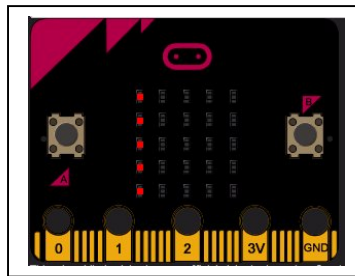
```
mycode.py
1 from microbit import *
2 while True:
3     for x in range (5):
4         display.set_pixel(x,0,9)
5
```



Tester les programmes pour allumer une ligne ou une colonne.

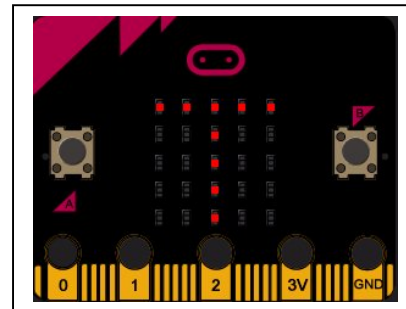
**Programme 6 :**

```
mycode.py
1 from microbit import *
2 while True:
3     for y in range (5):
4         display.set_pixel(0,y,9)
5
```



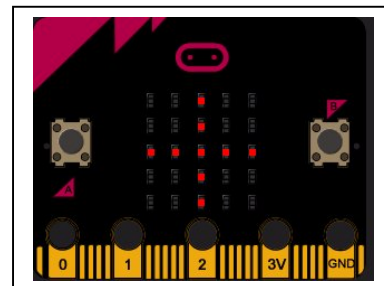
**Programme 7 :**

```
mycode.py
1 from microbit import *
2 while True:
3     for x in range (5):
4         display.set_pixel(x,0,9)
5     for y in range (5):
6         display.set_pixel(2,y,9)
7
8     sleep(500)
```



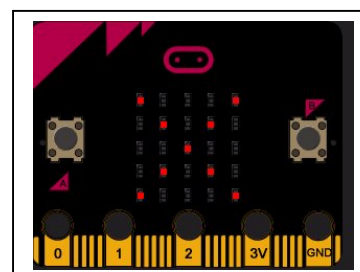
**Programme 8 :**

Modifier la ligne 4 du programme 7 pour obtenir l'affichage ci-contre



**Programme 9 :**

```
mycode.py
1 from microbit import *
2
3 while True:
4     x=-1
5     y=-1
6     for i in range(5):
7         x = x + 1
8         y = y + 1
9         display.set_pixel(x,y,9)
10        sleep(500)
11    x=5
12    y=-1
13    for i in range(5):
14        x =
15        y =
16        display.set_pixel(x,y,9)
17        sleep(500)
18
```



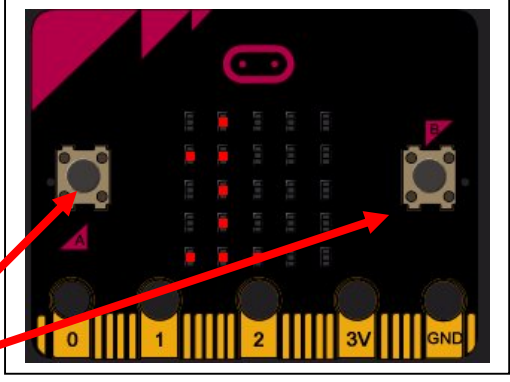
Compléter les lignes 14 et 15 du programme 9 pour afficher le X

**Programme 10 :**

```

1  from microbit import *
2
3
4  def carre(i, color):
5      for x in range(i):
6          for y in range(i):
7              display.set_pixel(x,y,color)
8
9  while True:
10     carre(1,9)
11     sleep(500)
12     carre(2,9)
13     sleep(500)
14     carre(3,9)
15     sleep(500)
16     carre(4,9)
17     sleep(500)
18     carre(5,9)
19     sleep(500)
20
21     # eteindre :
22     carre(5,0)
23     sleep(1000)
    
```

Tester le programme 10 .  
 Les appels successifs de la fonction carre peuvent être remplacés par une boucle  
 « for i in range (...) » :  
 Proposer une solution pour réécrire ce programme.  
 Note : on pourra utiliser display.clear() pour éteindre toutes les leds.



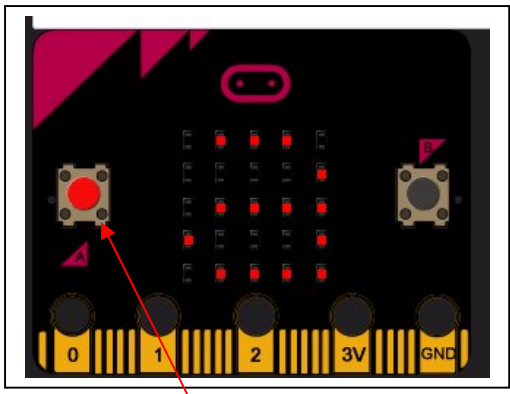
**Utilisation des boutons**

La carte possède deux boutons que l'on peut utiliser pour interagir avec le programme :

**Programme 11 :**

```

mycode.py
1  from microbit import *
2
3  while True:
4      if button_a.was_pressed():
5          display.show("a")
6          sleep(1000)
7          display.clear()
8      if button_b.was_pressed():
9          display.show("b")
10         sleep(1000)
11         display.clear()
    
```



Lancer le programme 11 puis cliquer sur les boutons gauche ou droite avec la souris

**Compléter :**

*if button\_a.was\_pressed(): permet de tester si le bouton ..... est appuyé*  
*if button\_b.was\_pressed(): permet de tester si le bouton ..... est appuyé*

**Programme 12 :**

```

1  from microbit import *
2
3  x=2
4  while True:
5      if button_b.was_pressed(): # si bouton appuyé
6          display.set_pixel(x,2,0) # efface position actuelle
7          x=x+1 # déplacer à droite la position
8          if x>4: # pour ne pas sortir de la matrice
9              x=4
10         display.set_pixel(x,2,9) # Afficher la nouvelle position
11         sleep(200) # temporiser
12
13     if button_a.was_pressed():
14         display.set_pixel(x,2,0)
15         x=x-1
16         if x<0:
17             x=0
18
19         display.set_pixel(x,2,9)
20         sleep(200)
21
    
```

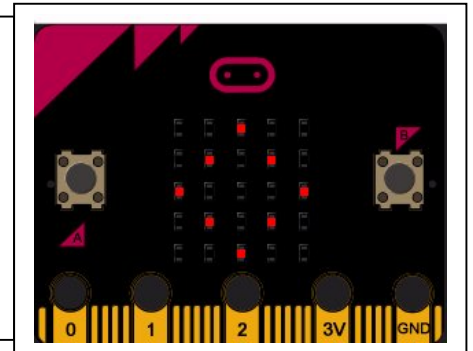
Tester le programme.  
 En cliquant sur les boutons, vous devriez voir les pixels s'éteindre et s'allumer successivement donnant l'illusion d'un déplacement.  
 A quoi servent les instructions des lignes 6 et 14 ?

**Programme 13 : Créer ses propres figures**

```

1 from microbit import *
2
3 img1 = [ Image('00900:09090:90009:09090:00900:') ]
4 img2 = [ Image('00000:09990:90009:09990:00000:') ]
5 while True:
6     display.show(img1)
7     sleep(1000)
8     display.show(img2)
9     sleep(1000)
10    display.clear()
11    sleep(500)

```



Img1 et img2 sont deux images : Sachant que le « 9 » représente la led allumée (intensité max) , expliquer à quoi correspond les 5 blocs de chiffres séparés par « : »

Faire un programme pour afficher un Z et son symétrique successivement.

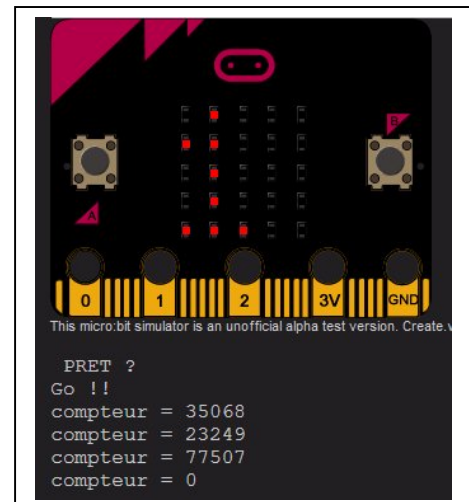
Faire un programme pour afficher un cœur qui clignote

**Programme 14 : Jeu de réflexe :**

```

1 from microbit import *
2
3 from random import randint
4 somme = 0
5 while True:
6     print (" PRET ?")
7     sleep (2000)
8     print ("Go !!")
9     somme = 0
10    for n in range (10):
11        a = randint(0,1)
12        if a==0:
13            display.show("1")
14            compteur = 0
15            while not(button_a.was_pressed()):
16                compteur = compteur+1
17            print ("compteur =", compteur)
18            display.clear()
19            sleep(500)
20            somme = somme + compteur
21
22        if a==1:
23            display.show("2")
24            compteur = 0
25            while not(button_b.was_pressed()):
26                compteur = compteur+1
27            print ("compteur =", compteur)
28            display.clear()
29            sleep(500)
30            somme = somme + compteur
31
32
33    print (" le plus petit score gagne ")
34    print (" score = ",somme)

```



Tester le programme.

Il faut appuyer (cliquer) sur le bouton gauche ou droite en fonction du chiffre affiché. Plus on est rapide, plus le score sera faible (le plus petit score gagne sur une série de 10 coups par joueur)