

---

# **Ressources pour la carte Micro:bit**

***Version 1***

**IREM Marseille**

**mai 09, 2018**



<b>1 Projets à réaliser</b>	<b>3</b>
1.1 Températures . . . . .	3
1.2 Coffre fort . . . . .	5
1.3 Boîte fermée . . . . .	6
1.4 Planche de Galton . . . . .	7
1.5 Pierrot et Simon . . . . .	8
<b>2 Index et page de recherche</b>	<b>11</b>
<b>Index</b>	<b>13</b>



par le groupe InEFLP de l'IREM de Marseille



# CHAPITRE 1

---

## Projets à réaliser

---

### 1.1 Températures

#### 1.1.1 Description

---

**À faire :** capture d'écran / gif animée.

---

#### Exemple(s) d'utilisation

##### Escape game

Nous avons utilisé le projet *Températures* pour un escape game proposé en stage.

- diaporama d'accueil : <http://url.univ-irem.fr/temp>
- page de formation : <http://url.univ-irem.fr/algo1718-temp>

#### 1.1.2 Réalisation

##### Fabriquer

Nous détaillons ici comment fabriquer et assembler le matériel nécessaire à la réalisation du projet *Températures*.

---

**À faire :** tout faire.

---

### Coder

Ce projet a été programmé en micropython.

Nous détaillons ici le code nécessaire à la réalisation du projet *Températures*.

### Le code, étape par étape

Incluons la bibliothèque Micro :bit

```
from microbit import *
```

### Le code final

```
1  # -*- coding: utf-8 -*-# Encoding cookie added by Mu Editor
2  from microbit import *
3
4  # définir mes images persos
5  # pour les lignes qui se colorent
6  image1 = Image(
7      '00000:'
8      '00000:'
9      '00000:'
10     '00000:'
11     '99999')
12  image2 = Image(
13     '00000:'
14     '00000:'
15     '00000:'
16     '99999:'
17     '77777')
18  image3 = Image(
19     '00000:'
20     '00000:'
21     '99999:'
22     '77777:'
23     '77777')
24  image4 = Image(
25     '00000:'
26     '99999:'
27     '77777:'
28     '77777:'
29     '77777')
30  image5 = Image(
31     '99999:'
32     '77777:'
33     '77777:'
34     '77777:'
35     '77777')
36
37  # booléen pour savoir si l'énigme est réussie
38  victoire = False
39
40  # à faire toujours et toujours...
```

(suite sur la page suivante)



(suite de la page précédente)

```

41 while True:
42     # si l'énigme n'a pas été résolue
43     if not victoire:
44         # lire la température (en °C)
45         temp = temperature()
46         # affichage des images en fonction
47         # de temp
48         if temp < 29:
49             display.clear()
50         elif 29 <= temp < 30:
51             display.show(image1)
52         elif 30 <= temp < 31:
53             display.show(image2)
54         elif 31 <= temp < 32:
55             display.show(image3)
56         elif 32 <= temp < 33:
57             display.show(image4)
58         elif 33 <= temp < 34:
59             display.show(image5)
60         # victoire !
61         elif 34 <= temp:
62             victoire = True
63             # petite animation
64             for i in range(2):
65                 display.show(Image.SQUARE_SMALL)
66                 sleep(100)
67                 display.show(Image.SQUARE)
68                 sleep(100)
69             sleep(500)
70
71     # si l'énigme est résolue
72     if victoire:
73         # petite image joyeuse
74         display.show(Image.HAPPY)
75         sleep(500)
76         # code secret à afficher...
77         display.scroll("XXXXXX")
78
79     # utiliser le bouton A pour réinitialiser
80     if button_a.is_pressed():
81         victoire = False

```

## 1.2 Coffre fort

### 1.2.1 Description

---

**À faire :** capture d'écran / gif animée

---

#### Exemple(s) d'utilisation

### Escape game

Nous avons utilisé le projet *Coffre fort* pour un escape game proposé en stage.

- diaporama d'accueil : <http://url.univ-irem.fr/coffre>
- page de formation : <http://url.univ-irem.fr/algo1718-coffre>

### 1.2.2 Réalisation

#### Fabriquer

Nous détaillons ici comment fabriquer et assembler le matériel nécessaire à la réalisation du projet *Coffre fort*.

---

**À faire :** tout faire.

---

#### Coder

Nous détaillons ici le code nécessaire à la réalisation du projet *Coffre fort*.

---

**À faire :** tout à faire !

---

## 1.3 Boîte fermée

### 1.3.1 Description

---

**À faire :** capture d'écran / gif animée

---

### Exemple(s) d'utilisation

#### Escape game

Nous avons utilisé le projet *Boîte fermée* pour un escape game proposé en stage.

- diaporama d'accueil : <http://url.univ-irem.fr/boite>
- page de formation : <http://url.univ-irem.fr/algo1718-boite>

### 1.3.2 Réalisation

#### Fabriquer

Nous détaillons ici comment fabriquer et assembler le matériel nécessaire à la réalisation du projet *Boîte fermée*.

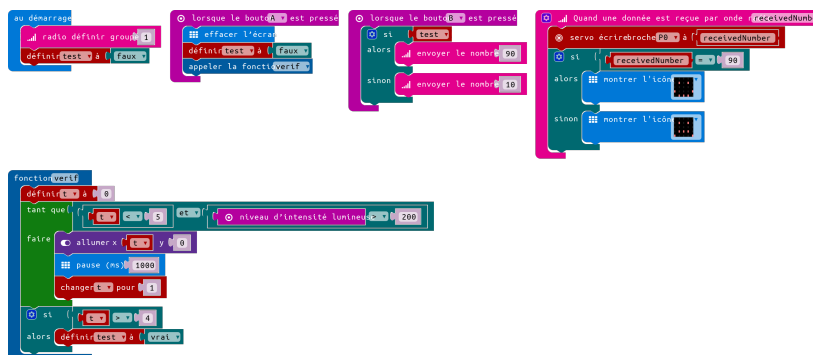
---

**À faire :** tout faire.

---

## Coder

Nous détaillons ici le code nécessaire à la réalisation du projet *Boîte fermée*.



## 1.4 Planche de Galton

### 1.4.1 Description

À faire : capture d'écran / gif animée

### Exemple(s) d'utilisation

### 1.4.2 Réalisation

#### Fabriquer

Nous détaillons ici comment fabriquer et assembler le matériel nécessaire à la réalisation du projet *Planche de Galton*.

À faire : tout faire.

## Coder

Nous détaillons ici le code nécessaire à la réalisation du projet *Planche de Galton*.

```
from microbit import *
from random import random, seed

seed(300)          # la graine de hasard ???
n = [0, 0, 0, 0, 0] # le tableau contenant les compteurs

def aff(n, m):      # la fonction affichant le graph
    q = n // 9       # nombre de led eclaire totalement
    r = n % 9        # portion de la derniere led eclaire
    for i in range(0, q):
```

(suite sur la page suivante)

```

        display.set_pixel(m, 4-i, 9)
        display.set_pixel(m, 4-q, r)

def chute(t):                                # fonction affichant la chute
    display.clear()
    y, x = 0, 0
    display.set_pixel(x, y, 9)
    sleep(t)
    while y < 4:
        display.clear()
        if round(random()):                  # si arrondi de alea est vrai (différent de 0)
            y = y + 1                        # on augmente y de 1
        else:
            x = x + 1
            y = y + 1
        display.set_pixel(x, y, 9)
        sleep(t)
    n[x] = n[x]+1                            # incrementation du compteur de la position x
    display.set_pixel(x, y, 1)

while True:
    if button_a.is_pressed():
        chute(500)

    elif button_b.get_presses():
        n = [0, 0, 0, 0, 0]
        for k in range(80):
            chute(round(500 / (1.05**k)))
            for j in range(5):
                aff(n[j], j)
            sleep(200)
        print(n)

```

## 1.5 Pierrot et Simon

### 1.5.1 Description

---

**À faire :** capture d'écran / gif animée

---

#### Exemple(s) d'utilisation

##### Escape game

Nous avons utilisé le projet *Pierrot et Simon* pour un escape game proposé en stage.

- diaporama d'accueil : <http://url.univ-irem.fr/pierrot>
- page de formation : <http://url.univ-irem.fr/algo1718-pierrot>

## 1.5.2 Réalisation

### Fabriquer

Nous détaillons ici comment fabriquer et assembler le matériel nécessaire à la réalisation du projet *Pierrot et Simon*.

---

**À faire :** tout faire.

---

### Coder

Nous détaillons ici le code nécessaire à la réalisation du projet *Pierrot et Simon*.

---

**À faire :** tout à faire !

---



## CHAPITRE 2

---

### Index et page de recherche

---

- [genindex](#)
  - [search](#)
-





## G

galton, [7](#)

## M

micropython, [4](#), [6](#), [7](#)

## P

projet, [4](#), [6](#), [7](#)

python, *voir* micropython