

# Pile ou face communicant avec Micro:bit (et python)

## DESCRIPTION

### Objectif

Le but de ce projet est de simuler une expérience aléatoire de lancer de pièce truquée avec une carte Micro:bit, et d'envoyer par radio l'issue afin de centraliser les résultats. Cette activité permet de prendre en main facilement l'interface de développement en python

### Intérêt

Cette activité, idéale pour découvrir l'interface de programmation, propose d'utiliser python pour programmer une situation non-équiprobable et d'exploiter le module radio du Micro:bit.

**Production et traitement de données** Cette situation permet de travailler des notions de probabilités et de statistiques.

**Programmation Python** Cette activité nécessite que les élèves aient déjà été initié à Python, sans pour autant demandé une grande maîtrise.

### Matériel



- 1 × Micro:bit
- 1 × IDE programmation python (Mu) <https://codewith.mu/> ou interface de programmation ligne <https://python.microbit.org/v/1.1>

### Progression proposée

#### MÉTHODE

On propose ici d'aborder la problématique en deux temps :

#### 1) Tirage aléatoire et envoi des issues

Les élèves doivent modéliser une expérience aléatoire avec une probabilité donnée, et envoyer l'issue par radio.

#### 2) Réception et traitement des données

L'incrémentation des variables et le traitement des données est déclenchée par la réception d'un message radio.

## NIVEAU SIMPLE

### Activité élève



Durée

0,5 h

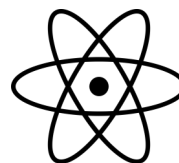


Public

2de



Maths

expérience  
aléatoire

Sciences



Algo

fonction ;  
événement ;  
boucle

#### ACTIVITÉ

MISSION : PROGRAMME MICRO:BIT POUR SIMULER UN LANCER DE **Pile ou Face** TRUQUÉ !

On veut que notre **pièce virtuelle** ait 55% de chance de tomber sur "pile".

Le programme doit débuter avec les instructions ci-dessous afin :

- 1) d'importer les bibliothèques nécessaires ;

```
from microbit import *  
import random  
import radio
```

- 2) allumer le mode radio ; `radio.on()`

- 3) lancer une boucle infinie et détecter si le Micro:bit a été secoué ;

```
while True:  
    if accelerometer.was_gesture('shake'):
```

Il faut maintenant ajouter les fonctions pour :

- 1) effectuer un tirage aléatoire d'un nombre entre 1 et 100 avec `random.randint(,)` et comparer ce nombre à la probabilité voulue.
- 2) afficher P ou F **selon l'issue du tirage** avec `display.show('')` ;
- 3) envoyer la valeur 'P' ou 'F' par radio **selon l'issue du tirage** avec `radio.send('')`

Vérifie ton code avec



et flash-le sur la carte avec





## Notes pour l'enseignant

Ce premier niveau permet de se familiariser avec l'interface de l'IDE en produisant un premier programme fonctionnel et utile. Les notions de programmations utilisées sont :

- 1) l'appel de fonction
- 2) la boucle `while`
- 3) la boucle `if`

Dans cette première partie l'élève est aussi amené à concevoir une expérience aléatoire avec une probabilité donnée.

### MÉTHODE

Pour résoudre ce problème, il suffit de programmer les instructions de la façon suivante :

```
from microbit import *
import random
import radio

radio.on()

while True:
    if accelerometer.was_gesture('shake'):
        tirage = random.randint(1, 100)

        if tirage <= 55:
            display.show("P")
            radio.send('P')
        else:
            display.show("F")
            radio.send('F')
        sleep(10)
```

### REMARQUE

La variable `tirage` n'est pas indispensable, elle permet d'enregistrer le résultat du tirage aléatoire et de rendre le code plus lisible.

L'instruction `sleep(10)` permet de faire une pause de 10 ms dans le programme.

Afin de distinguer deux tirages successifs, on peut marquer une pause puis effacer l'écran à l'issue d'un tirage avec `display.clear()`, mais il peut-être amusant de montrer comment faire une animation simplement à partir d'une succession d'images.

Par exemple la fonction `anim()` ci-dessous pourra être définie au début du programme puis appeler avant chaque tirage.

```
while True:
    if accelerometer.was_gesture('shake'):
        tirage = random.randint(1, 100)
        anim()
```

# NIVEAU EXPERT

## Activité élève



Durée

0,5 h



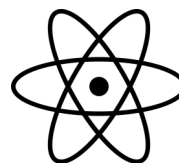
Public

2de



Maths

fréquences



Sciences



Algo

fonction ;  
événement ;  
variable


### ACTIVITÉ

MISSION : RÉCEPTIONNE LES DONNÉES ENVOYÉES PAR LES AUTRES MICRO:BIT ET EXPLOITE-LES POUR TRACER UN **graphique**

Dans l'activité précédente nous avons créer une **pièce virtuelle** qui affiche 'P' ou 'F' et qui envoie cette information par radio. Nous allons programmer le Micro:bit afin de recevoir, traiter et afficher les données.

Le programme précédent devra être complété avec les éléments suivants :

- 1) une déclaration de variables pour dénombrer les 'P' et les 'F', comme par exemple `p = 0` , qu'il faut écrire **avant** la boucle `while True:`
- 2) la réception d'un message radio (que l'on pourra enregistrer dans une variable 'issue') avec la fonction `radio.receive()` ;
- 3) l'incréméntation des variables décomptant les 'P' et les 'F' par exemple `p = p + 1` ;
- 4) le calcul de fréquence de 'P' et de 'F' (attention il faudra veiller à ne pas diviser par 0);
- 5) l'affichage des fréquences dans le plotter avec `print((a,b))` où '(a,b)' est le couple de fréquences de 'P' et de 'F' dont on souhaite voir l'évolution au cours du temps.

Vérifie ton code avec



et flashe-le sur la carte avec



Avant d'afficher le plotter avec



, il faudra peut-être faire un reset de la carte.

## Notes pour l'enseignant

Cette deuxième activité nécessite impérativement l'usage de variables.

La difficulté peut provenir des différents cas de figure à traiter, mais à part l'instruction `else:`, il n'y a pas de nouveaux types d'instruction.

Les notions de programmations utilisées sont :

- 1) la déclaration et l'initialisation de variables
- 2) la boucle `if`, complétée par `else`

Dans cette partie l'élève est aussi amené à calculer des fréquences, mais l'exécution de ce calcul doit être conditionné à la non nullité de la somme des variables.

### MÉTHODE

Pour recevoir et exploiter les données, il est possible de programmer les instructions de la façon suivante :

```
from microbit import *
import random
import radio

radio.on()

f = 0
p = 0

def anim():
    tab = [Image.ARROW_N, Image.ARROW_NE, Image.ARROW_E, Image.ARROW_SE,
           Image.ARROW_S, Image.ARROW_SW, Image.ARROW_W, Image.ARROW_NW]
    display.show(tab, delay=80)

while True:
    if accelerometer.was_gesture('shake'):
        tirage = random.randint(1, 100)
        anim()
        if tirage <= 55:
            display.show("P")
            radio.send('P')
        else:
            display.show("F")
            radio.send('F')
        sleep(10)

    issue = radio.receive()
    if issue == 'P':
        p += 1
    if issue == 'F':
        f += 1
    if p+f > 0:
        a = p/(p+f)
        b = f/(p+f)
    else:
        a, b = 0, 0
    print((a, b))
    sleep(10)
```

---

# À propos de cette publication

## POURQUOI LES OBJETS CONNECTÉS ?

Alors que dans certaines disciplines le temps commence à manquer pour traiter l'ensemble du programme, certains évoquent déjà l'idée d'en faire plus !

En effet, les enseignants utilisent déjà les outils numériques. Par exemple, dans les classes de mathématiques, l'utilité du tableur et de GeoGebra n'est plus à démontrer. Jusqu'à l'introduction de l'algorithmique, ces deux logiciels efficaces et maîtrisés par les enseignants étaient amplement suffisants. Est-ce donc juste un effet de mode de faire cours avec les robots (Thymio, Mbot), les objets programmables et connectés (Arduino, Micro:bit, STM education, Raspberry Pi) ou est-ce une nouvelle façon d'aborder notre enseignement ? Ces nouvelles possibilités technologiques, forcément chronophages, nous permettront-elles de traiter un contenu disciplinaire exigeant dans un cadre institutionnel contraignant ?

Nous n'avons bien sûr pas toutes les réponses à ces questions mais nous pensons que lorsqu'il est accompagné de certains de ces outils, notre enseignement a beaucoup à y gagner.

L'introduction de l'algorithmique en lycée professionnel nous interroge. Longtemps il nous a semblé impensable et inenvisageable d'avoir à enseigner un langage de programmation comme Python auprès d'un public d'élèves globalement en difficulté avec les mathématiques. Fort de ce constat, nous avons cherché les moyens de lier les mathématiques à la logique et au raisonnement algorithmique. C'est pourquoi nous avons exploré les potentialités des objets connectés.

Notre postulat est double. Nous pensons que :

- grâce à des situations réelles et concrètes, les objets connectés facilitent la mise en activité de tous les élèves ;
- grâce à des activités simples mais évolutives centrées autour de réalisations matérielles, la dimension affective du travail est valorisée. Soyons fous et espérons que l'élève tisse une histoire personnelle avec l'activité, qu'il soit fier du travail accompli et qu'il prenne également du plaisir à expliquer et à montrer ses réalisations.

En devenant de plus en plus simples, accessibles et facilement utilisables, les objets connectés permettent d'aborder des contenus disciplinaires et de développer des compétences transversales essentielles pour l'élève.

En travaillant à partir des objets connectés, la situation de départ est plus concrète et l'objectif à atteindre suffisamment clair pour l'élève. Plus ou moins guidé selon son niveau d'expertise technique, il est alors libre dans sa démarche. Avec des interfaces de programmation accompagnées parfois de simulateurs, la démarche par essais et erreurs a ici toute sa place. Par ailleurs, l'élève devra clarifier sa pensée avant de verbaliser ses idées en langage naturel. Il pourra ainsi proposer et élaborer un modèle acceptable par la machine pour enfin traduire son algorithme en se pliant à la rigueur du langage de programmation.

Effectuant régulièrement des va-et-vient entre abstraction et réalité, cherchant à valider son algorithme à partir d'un visuel ou d'une exploitation des résultats, l'élève entre progressivement dans la modélisation.

Les scénarios proposés dans cette brochure permettent tout cela : une approche des mathématiques et des sciences qui laisse la place à l'expérimentation : manipulation, programmation et auto-validation.

## QUI SOMMES-NOUS ?

Nous sommes des enseignants de maths/sciences regroupés au sein d'un groupe de recherche de l'IREM de Marseille.



Notre groupe, Innovation, Expérimentation et Formation en Lycée Professionnel (InEFLP) consacre une partie de son travail à l'enseignement de l'algorithmique en classes de lycée professionnel. Dans le cadre de cette recherche, nous explorons les objets connectés tels que Arduino, Micro:bit, STM32 Éducation ou mbot.

## LIENS UTILES

### Page du groupe InEFLP

<http://url.univ-irem.fr/ineflp>

### IREM de Marseille Site académique de l'IREM de Marseille

<http://url.univ-irem.fr/mars>

### Portail des IREM Site national des IREM

<http://www.univ-irem.fr/>

### Formation à l'algorithmique LP et SEGPA Padlet de utilisé lors de nos formations académiques

<http://url.univ-irem.fr/stage-algo>

### Collecte de ressources pour Micro:bit Padlet sur Micro:bit utilisé en formation

<http://url.univ-irem.fr/algo2017-microbit>

### Brochure sur Micro:bit Publication de la C2i TICE pour une prise en main de Micro:bit

<http://url.univ-irem.fr/c2it-mb-t1-pdf>

### Description Micro:bit Fiche sommaire de description de Micro:bit

<http://url.univ-irem.fr/ineflp-microbit>

### Site IREM dédié à Micro:bit Site de ressources sur Micro:bit du groupe

<http://url.univ-irem.fr/o>



*Un extrait de la brochure*

## Les objets connectés pour enseigner l'algorithmique en lycée professionnel

< version du 30 janvier 2020 >