

**Savoir-faire** : proposer et mettre au point un programme conformément au résultat attendu.

---

**Problématique** : comment programmer des jeux avec la carte micro:bit ?

## Première partie

La première partie consiste à programmer un dé électronique.

### 1. Dé électronique

Lors de l'appui sur un des boutons, la valeur du dé doit s'afficher sur la carte micro:bit. Puis au bout de quelques secondes, l'écran doit s'effacer.



Les instructions ci-dessous sont nécessaires pour implémenter le dé électronique.

#### 1.1. Choix aléatoire d'un chiffre

La bibliothèque `random` permet de choisir aléatoirement un chiffre.

L'instruction `import random`, qui permet d'importer la bibliothèque `random`, est à placer au début du programme (avant la boucle `while`).

L'instruction `randint(val1, val2)` donne un chiffre aléatoire compris entre `val1` et `val2`.

Exemple :

```
val = random.randint(1, 4)
```

#### 1.2. Effacer l'écran

L'instruction `display.clear()` permet d'effacer l'écran.

#### 1.3. Implémentation du programme du dé électronique

**Q1** : Compléter le programme du dé électronique.

```
from microbit import *  
import random  
  
while True:
```

## 2. Variante du dé électronique

La variante consiste à afficher les faces du dé et non les chiffres.



### 2.1. Création d'une image

La création d'une image s'effectue grâce à l'instruction `Image()`.

Par exemple, l'instruction ci-dessous dessine le carré ci-dessous.

```
carre = Image(" 99999:
90009:
90009:
90009:
99999:")
```

Le chiffre 9 permet d'allumer la led et le chiffre 0 permet de l'éteindre.  
Chaque nombre (par exemple : 9999) représente une ligne.

Pour simplifier l'écriture de l'instruction ci-dessus, il est possible de l'écrire comme ci-dessous :

```
carre = Image("99999:90009:90009:90009:99999:")
```

### 2.2. Choix aléatoire entre des images

Le nom des images doit être stocké dans une liste.

L'instruction `choice` permet de la bibliothèque `random` permet de choisir aléatoirement un élément de la liste.

Dans l'exemple qui suit, l'affichage du dessin correspondant aux chiffres un et deux se fait aléatoirement.

```
un = Image("00000:00000:00900:00000:00000:")
deux = Image("00000:00090:00000:09000:00000:")
```

```
liste_image = [un, deux]
```

```
display.show(random.choice(liste_image))
```

**Q2** : Compléter le programme de la variante du dé électronique.

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

```
un = Image("00000:00000:00900:00000:00000:")
deux =
```

```
liste_image =
```

## Deuxième partie

La première partie consiste à programmer un chifoumi.

### 3. Chifoumi

- Le lancement du chifoumi s'effectue lorsque la carte est secouée.

#### 3.1. Images du chifoumi

Les images ci-dessous sont utilisés.



pierre



feuille



ciseaux

#### 3.2. Détection du mouvement de la carte

La carte micro:bit possède un accéléromètre qui permet de détecter les mouvements de la carte. L'instruction `accelerometer.was_gesture("shake")` retourne `True` lorsque la carte est secouée.

**Q3** : À l'aide des informations ci-dessus, implémenter un chifoumi :