

Document support de l'activité "Bal de fin d'année"

Contexte : Cette activité s'inscrit au cœur de la séquence traitant de l'émission et de la perception d'un son. Toute la séquence met en scène un village dans lequel se déroule une fête, qui permet de faire intervenir toutes les notions de cette partie du programme. L'activité clôt la séquence et reprend les notions déjà étudiées, en y ajoutant l'utilisation du microcontrôleur.

Le microcontrôleur utilisé pour cette activité est la carte Microbit, qui est facile à prendre en main et permet de coder directement en Python. La majorité de ce qui suit peut être trouvé sur le site BBC microbit sur ce lien : <https://microbit-micropython.readthedocs.io/fr/latest/tutorials/music.html>

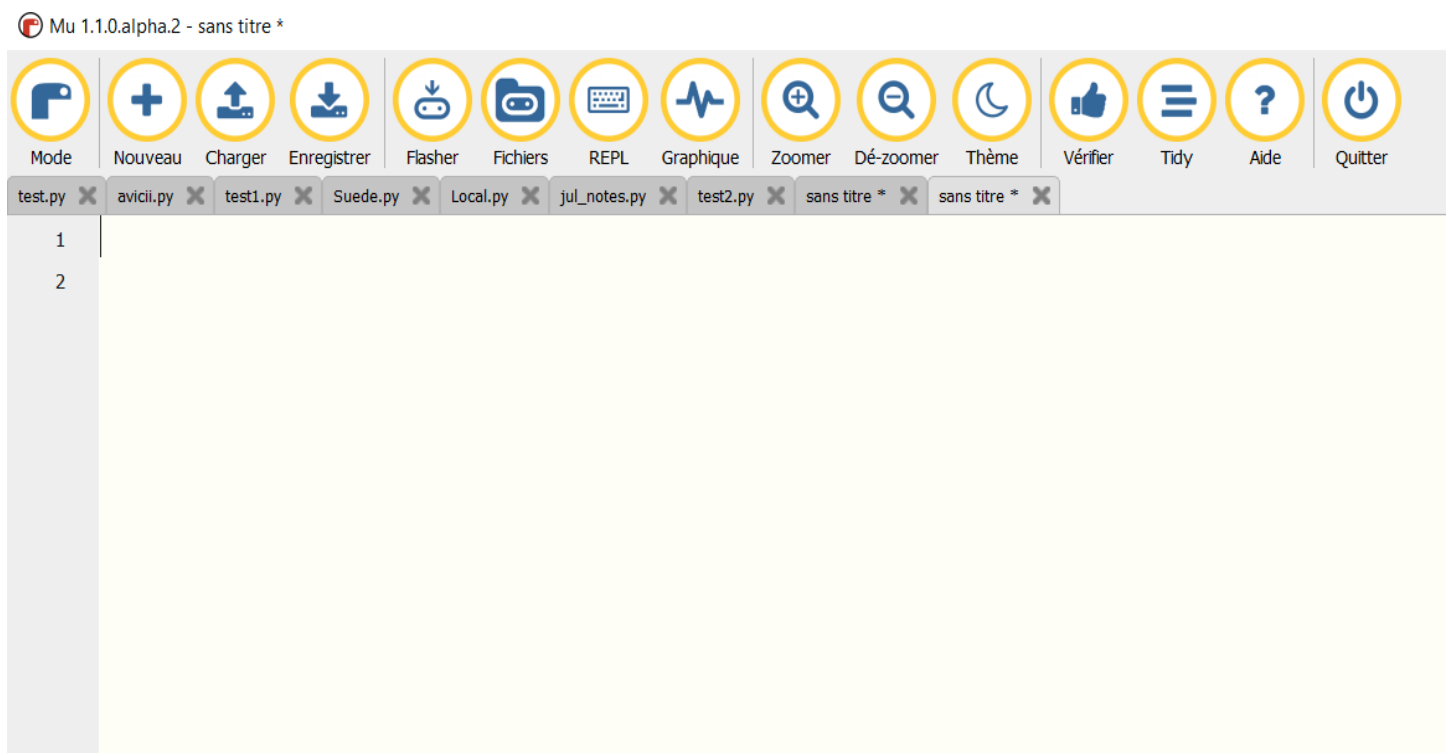
Le site est très bien documenté et présente les informations de manière progressive.

Prérequis : Physique Chimie : Les élèves doivent avoir vu les notions de fréquence et de période.

SNT : Les élèves doivent avoir étudié le microcontrôleur. Il peut être pertinent de s'entendre avec le professeur de SNT pour que la notion soit abordée juste avant cette séance qui permettra aux élèves de sécuriser leurs acquis.

I- La carte MicroBit et les bibliothèques musicales

La carte MicroBit est directement branchée en USB sur l'ordinateur. On peut y intégrer un programme directement à l'aide de l'éditeur Mu, qui se présente comme suit :



Une fois le programme écrit dans l'éditeur, il suffit d'appuyer sur le bouton "Flasher" pour que le programme soit transféré sur la carte, qui se mettra, dès qu'il sera chargé, à l'exécuter. Rappelons que la carte MicroBit ne peut avoir qu'un seul programme de chargé à la fois.

La carte MicroBit a dans sa banque la bibliothèque music, qui possède de nombreuses fonctionnalités. Il faut bien penser à l'inclure en début de programme à l'aide de la commande **'import music'**

On peut notamment générer un son en utilisant deux possibilités.

I- Première méthode (physique)

La première, la plus simple, consiste simplement à utiliser la commande 'music.pitch(X,Y)'. X est la fréquence du son, en Hz. Y est la durée du son que l'on souhaite générer, en ms.

```
1 from microbit import *
2 import music
3
4 music.pitch(440,500)
5 music.pitch(350,1000)
```

Par exemple, la commande 'music.pitch(440,500)' générera une note de fréquence 440 Hz (un la) durant 500 ms.

Une fois la note générée, le programme passera à la ligne suivante, et enchaînera donc une note de fréquence 350 Hz durant 1000 ms.

II- Deuxième méthode (musicale)

La deuxième méthode, plus proche de la musique, consiste à définir d'abord une mélodie, puis la générer.

```
1 from microbit import *
2 import music
3
4 toto=["A4", "D4"]
5 music.play(toto)
6
```

On voit qu'ici on définit d'abord la mélodie 'toto' puis on la joue à l'aide de la commande 'music.play(toto)'.

1- Définition des notes

Pour définir la mélodie on utilise la notation américaine, où chaque note de la gamme (do, ré, mi, fa, sol, la, si) correspond à une lettre. Le La correspond au A, et ainsi de suite. Pour se repérer dans les octaves, on accole un chiffre à la lettre. A4 est donc le 'la' de la quatrième octave. Si on souhaite mettre une pause, on peut introduire la « note » 'R' qui correspond à une pause.

2- Définition de la durée

Par défaut, toutes les notes font la même durée arbitraire, appelée tick.

La ligne de code : `toto=["A4"]`

est strictement équivalente à `toto=["A4:1"]`

Toutes les durées seront définies par des chiffres, correspondant au nombre de ticks que durera la note.

```
toto=["A4:1", "D4:2", "E4:4", "G4:3"]
```

Dans l'exemple, nous aurons un A4 d'une durée arbitraire de 1 tick, un D4 durant 2 fois la durée du A4, un E4 durant 4 fois celle du A4, et un G4 durant 3 fois celle du A4, et ainsi de suite.

3- Pour aller plus loin

On peut modifier la durée du tick à l'aide de la commande `music.set_tempo(ticks=X, bpm=Y)` avec X prenant un chiffre en entrée et définira la pulsation de référence (par exemple 3 ticks, 4 ticks) et Y prenant un nombre en entrée, définissant le nombre de pulsation par minutes.

```
music.set_tempo(ticks=4, bpm=145)
```

Ici, notre pulsation de référence dure 4 ticks, et nous aurons 145 fois cette référence dans une minute.

```
music.set_tempo(ticks=2, bpm=120)
```

Ici, notre pulsation de référence dure 2 ticks, et nous avons 120 pulsations par minutes.

4- Simplifications

Pour simplifier le code, MicroBit permet d'éviter certaines redondances. On pourra simplifier la ligne ci-dessous :

```
toto=["A4:1", "D4:4", "E4:4", "G4:3", "D5:4", "E5:4", "R:4", "C4:3"]
```

La première simplification concerne les octaves. Si on a une série de notes de la même octave, il suffit de définir l'octave à la première note de la série comme suit :

```
toto=["A4:1", "D:4", "E:4", "G:3", "D5:4", "E:4", "R:4", "C4:3"]
```

On voit qu'ici, on ne redéfinit l'octave que lors des changements.

La deuxième simplification concerne les durées. Si deux notes consécutives ont la même durée, on ne définit que la durée de la première :

```
toto=["A4:1", "D:4", "E", "G:3", "D5:4", "E", "R", "C4:3"]
```

Le code s'en trouve donc allégé. Notons la présence de la pause 'R', qui ici durera 4 ticks.

II- L'activité

L'activité repose sur plusieurs ressources :

- Le sujet de l'activité
- 4 programmes MicroBit :
 - test1 : programme permettant la prise en main de la première méthode de génération d'un son.
 - test2 : programme permettant la prise en main de la deuxième méthode de génération d'un son.
 - Local : programme jouant une mélodie modifiée qu'il faut "réparer" en utilisant la première méthode. Le morceau est un extrait de "L'Ovni" de Jul.
 - Suede : programme jouant une mélodie modifiée utilisant la deuxième méthode. Le morceau est un extrait de "Wake me up" d'Avicii.

Il est demandé à l'élève à l'aide des deux programmes tests de modifier une note à l'aide de chacune des méthodes, puis de jouer deux notes à l'aide des deux méthodes.

Une fois la prise en main effectuée, on lui donne les représentations des notes à modifier dans Suede et Local. A l'aide des tableaux fournis, il identifie les notes et les modifie dans les programmes afin qu'ils jouent les bonnes mélodies.