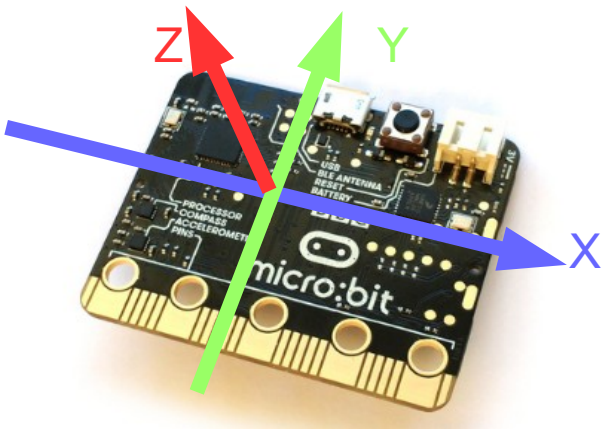
Partie 2 : Micro:bit - Utilisation de l'accéléromètre et de la boussole.

La carte BBC micro:bit est équipée d'un circuit accéléromètre MMA8652 produit par la société NXP/Freescale. Il permet d'obtenir une mesure de l'accélération suivant les trois axes X, Y, Z, positionnés comme sur la figure suivante :



Les valeurs sont accessibles dans le code en utilisant les fonctions Python suivantes :
acceleration.get_x() : pour obtenir la valeur de l'accélération suivant l'axe X
acceleration.get_y(): pour obtenir la valeur de l'accélération suivant l'axe Y
acceleration.get_z(): pour obtenir la valeur de l'accélération suivant l'axe Z
acceleration.get_values() : pour obtenir les 3 valeurs dans un tuple (accX , accY, accZ)

La valeur renvoyée par ces 3 fonctions est l'accélération en mg (milli-g), on a l'équivalence suivante 1 g = 1000 mg.
Le 'g' est l'unité d'accélération utilisée dans l'industrie de l'aéronautique, ou l'automobile par exemple. En physique l'unité du système international pour l'accélération est le mètre par seconde carré ($m.s^{-2}$). On a la correspondance suivante 1 g = 9,81 $m.s^{-2}$, il correspond à l'accélération de la pesanteur utilisée dans la célèbre formule P = m.g !

Manipulation 1

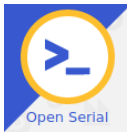
Dans l'éditeur python MU, saisir le code suivant :

```
from microbit import *

while True:
    a_x = accelerometer.get_x() #lecture et memorisation de l'accelerations suivant l'axe X
    a_y = accelerometer.get_y() #lecture et memorisation de l'accelerations suivant l'axe Y
    a_z = accelerometer.get_z() #lecture et memorisation de l'accelerations suivant l'axe Z

    print("accelerations : " , a_x, a_y, a_z) # affichage des valeurs dans REPL
    sleep(500) # temporisation nécessaire entre 2 lectures sur l'accéléromètre
```

Sauvegarder le code, le télécharger dans la micro:bit, et ouvrir *open serial* pour lire les valeurs.



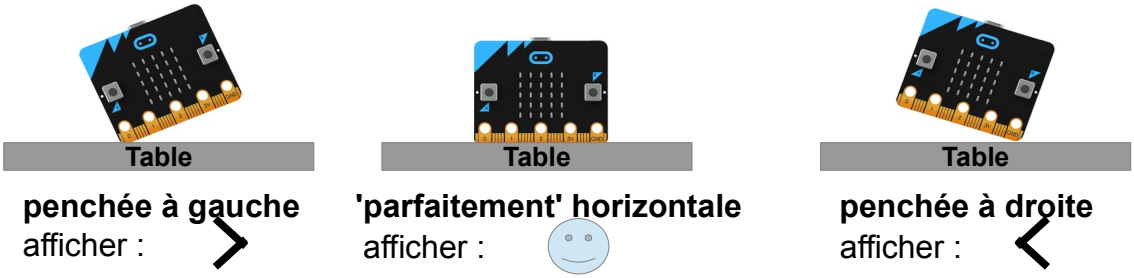
Mettre la carte à plat sur la table, observer les valeurs.
1. Quelle est la valeur la plus grande parmi les 3 composantes affichées ?
2. Convertir cette valeur en $m.s^{-2}$?
3. Sur quel axe se porte cette accélération ?
4. A quel phénomène physique, correspond cette valeur ?
5. Modifier la position de la carte (la mettre sur la tranche par exemple).
6. Comment évolue les valeurs affichées ?
7. Observer en particulier sur quel axe (X, Y ou Z) se trouve la valeur maximale précédente.
8. Cela vous parait-il cohérent ? Expliquer.

Mini-projet Réalisation d'un niveau à bulle

Un niveau à bulle est un instrument utilisé par les maçons pour vérifier l'horizontalité d'un mur.

Programmer la carte micro:bit afin qu'elle serve de niveau à bulle en utilisant le bord de l'axe X comme référence.

La carte posée parfaitement horizontale suivant l'axe X sur la table devra afficher le symbole SMILE
La carte positionnée inclinée vers la gauche devra afficher sur les LEDs le symbole ARROW-W
La carte positionnée inclinée vers la droite devra afficher sur les LEDs le symbole ARROW-E



RAPPEL
Voici les lignes de codes pour afficher les symboles sur la matrice de LEDs,

- la flèche orientée vers la gauche : `display.show(Image.ARROW_W)`
- la flèche orientée vers la droite : `display.show(Image.ARROW_E)`
- le smiley : `display.show(Image.SMILE)`

Implémenter le programme en code Python et le télécharger dans la carte micro:bit.
Tester physiquement votre niveau à bulle. Est-il pratique d'utilisation ? Proposer et implémenter une amélioration pour le rendre plus pratique d'utilisation.

<u>Manipulation 2 :</u> Utilisation de la détection de mouvement
Des algorithmes de détection de mouvement sont implémentés dans le micro:bit, ils s'appuient sur les valeurs captées par l'accéléromètre.
Ainsi la carte reconnaît les gestes suivant :

Up : la carte orientée vers le haut
Down : carte orientée vers le bas
Right : carte orientée vers la droite
Left : carte orientée vers la gauche,
Face Up : face de la carte vers le haut (position)
Face Down : face de la carte vers le bas (position)
Freefall : carte en chute libre
Shake : carte secouée

La fonction **accelerometer.current_gesture()** renvoie une chaîne de caractères (un mot) correspondant à la position de la carte conformément aux possibilités ci-dessus.
Le code suivant permet de tester les différentes possibilités :

```
from microbit import *
mvt_dernier = "" # mémorise le dernier mouvement fait
while True:
    mvt_actuel = accelerometer.current_gesture() # lecture du mouvement en cours
    sleep(100)
    if mvt_actuel is not mvt_dernier: # si le mouvement a changé depuis la dernière fois
        mvt_dernier = mvt_actuel      # on le mémorise
        print('{g:s}'.format(g=mvt_actuel)) # et on l'affiche dans REPL
```

Saisir le code, le sauvegarder et le flasher dans la micro:bit. Ouvrir *open serial* et tester les différentes positions de la carte.

Mini-projet ChiFouMi (Pierre – Papier – Ciseau)

règle du ChiFouMi
Il faut au moins deux joueurs. Les joueurs se font faces et mettent la main participante derrière le dos, le temps de prononcer la formule CHI – FOU – MI et chacun montre le choix qu'il a fait entre la pierre (main en poing), le papier (main à plat), ou le ciseau (doigt en forme de ciseau) .

- les ciseaux coupent la feuille
- la pierre brise les ciseaux
- la feuille recouvre la pierre

Le gagnant est désigné lorsque son choix correspond à une des règles ci-dessus, le match est nul sinon. Plusieurs manches peuvent être faites.

Créer une application qui permet de jouer au ChifouMi avec la carte micro:bit. Lorsque la carte détecte qu'elle est secouée, elle affiche le symbole de la pierre, du papier ou ciseau que l'algorithme aura choisi de manière aléatoire.

A l'aide de l'éditeur, ouvrir le fichier chiFouMi.py.

Les symboles graphiques correspondant à la PIERRE, au PAPIER, et au CISEAU, sont déjà définis dans le code.
L'affichage d'un symbole se fait par l'instruction :
display.show(PIERRE) *# pour afficher la pierre par exemple*

La fonction **accelerometer.is_gesture('shake')** renvoie **True** lorsque l'accéléromètre détecte que la carte est secouée.

Proposer au brouillon un algorithme, et coder / flasher le dans la carte micro :bit à l'aide de l'éditeur .