



# Comment vivons-nous la situation actuelle ?



# Comment vivons-nous la situation actuelle ?



# Activité de fitness avec les haltères ?

Il nous faudrait un accompagnement sportif personnel pour mesurer les exercices réalisés par séance et contribuer à notre motivation.



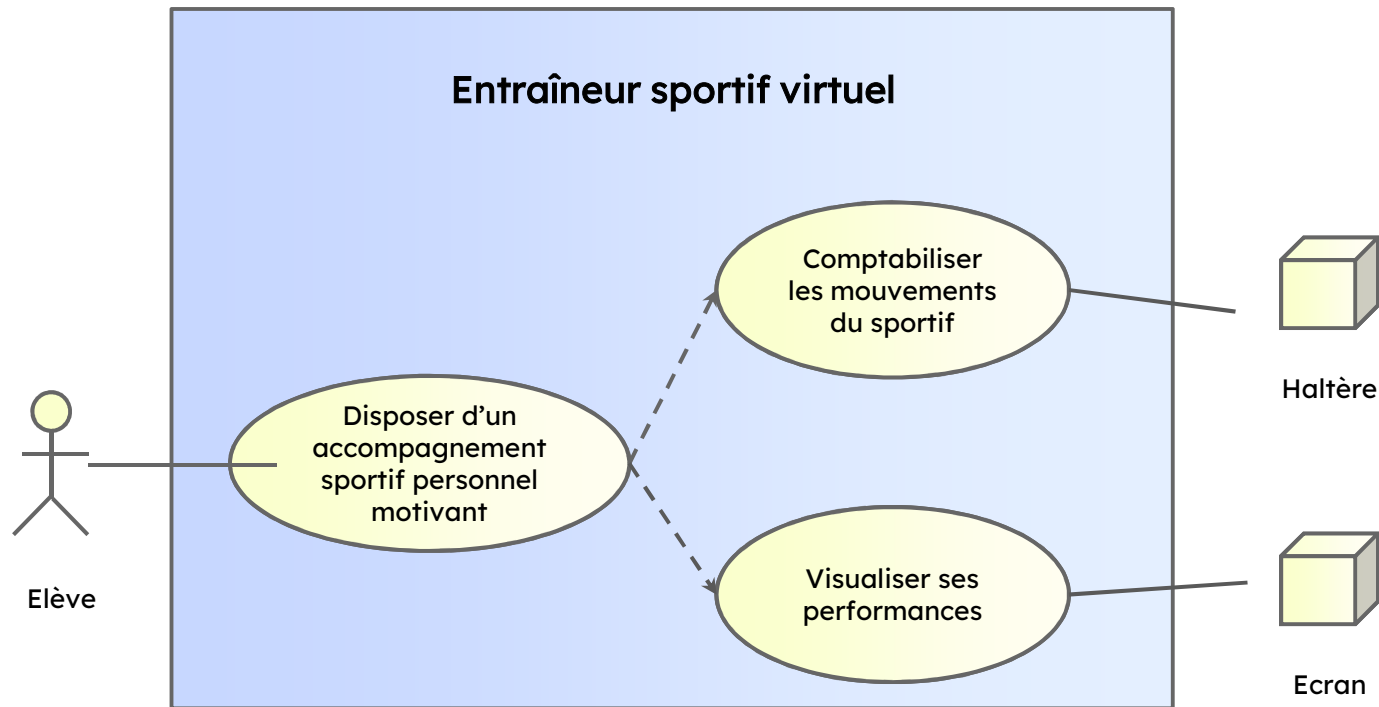
L'OMS considère que la sédentarité est déconseillée pour rester en bonne santé et préconise la pratique régulière du sport.

Comment peut-on, à la maison, visualiser le nombre de mouvements sportifs effectués en restant motivé ?



<https://www.who.int/dietphysicalactivity/publications/9789241599979/fr/>

# Comment peut-on, à la maison, visualiser le nombre de mouvements sportifs effectués en restant motivé ?



# Représentation du flux d'information

Nombre de mouvements



Chaîne d'information

Acquérir

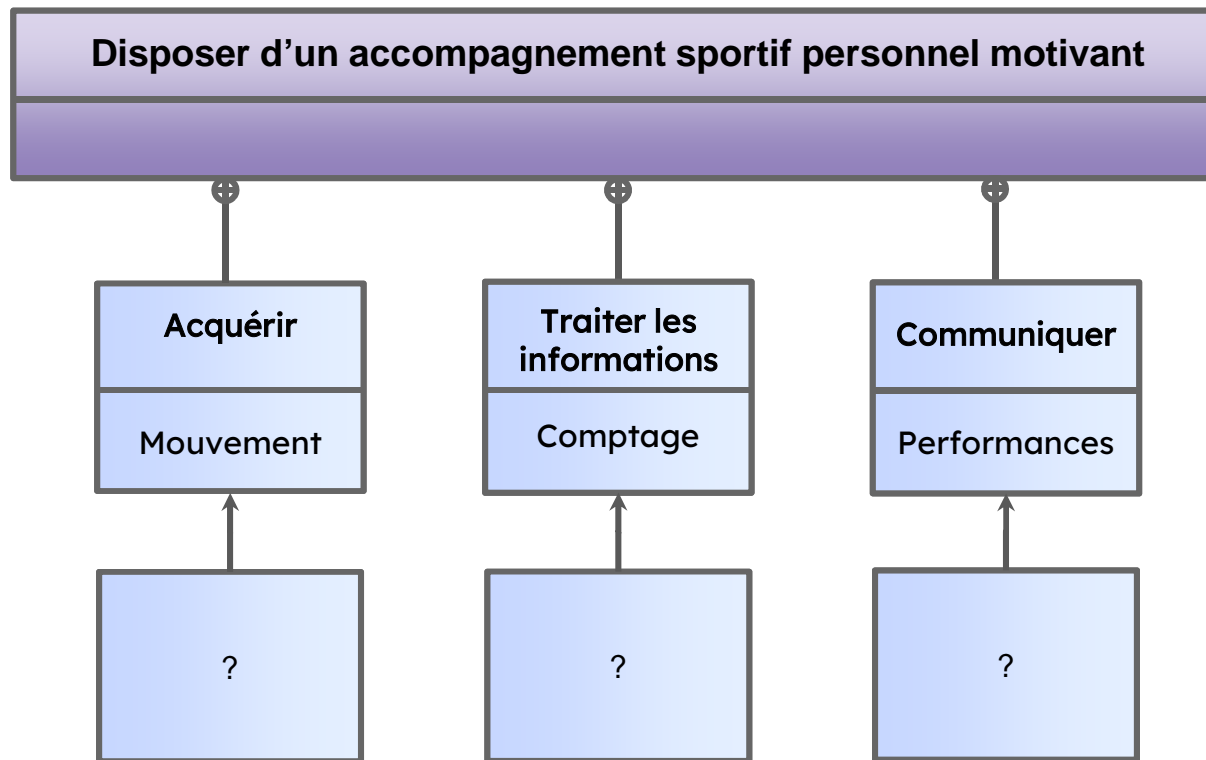
Traiter

Communiquer

Affichage des performances



# Représentation du cahier des charges fonctionnel :





# Points du programme

## L'informatique et la programmation

Attendus de fin de cycle

Ecrire, mettre au point et exécuter un programme

Connaissances et compétences associées

Ecrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.

Ecrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

- Notions d'algorithme et de programme
- Notion de variable informatique.
- Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles.
- Systèmes embarqués
- Capteur, actionneur, interface

# Points du programme

## La modélisation et la simulation des objets et systèmes techniques

Attendus de fin de cycle

Analyser le fonctionnement et la structure d'un objet

Connaissances et compétences associées

Analyser le fonctionnement le fonctionnement et la structure d'un objet

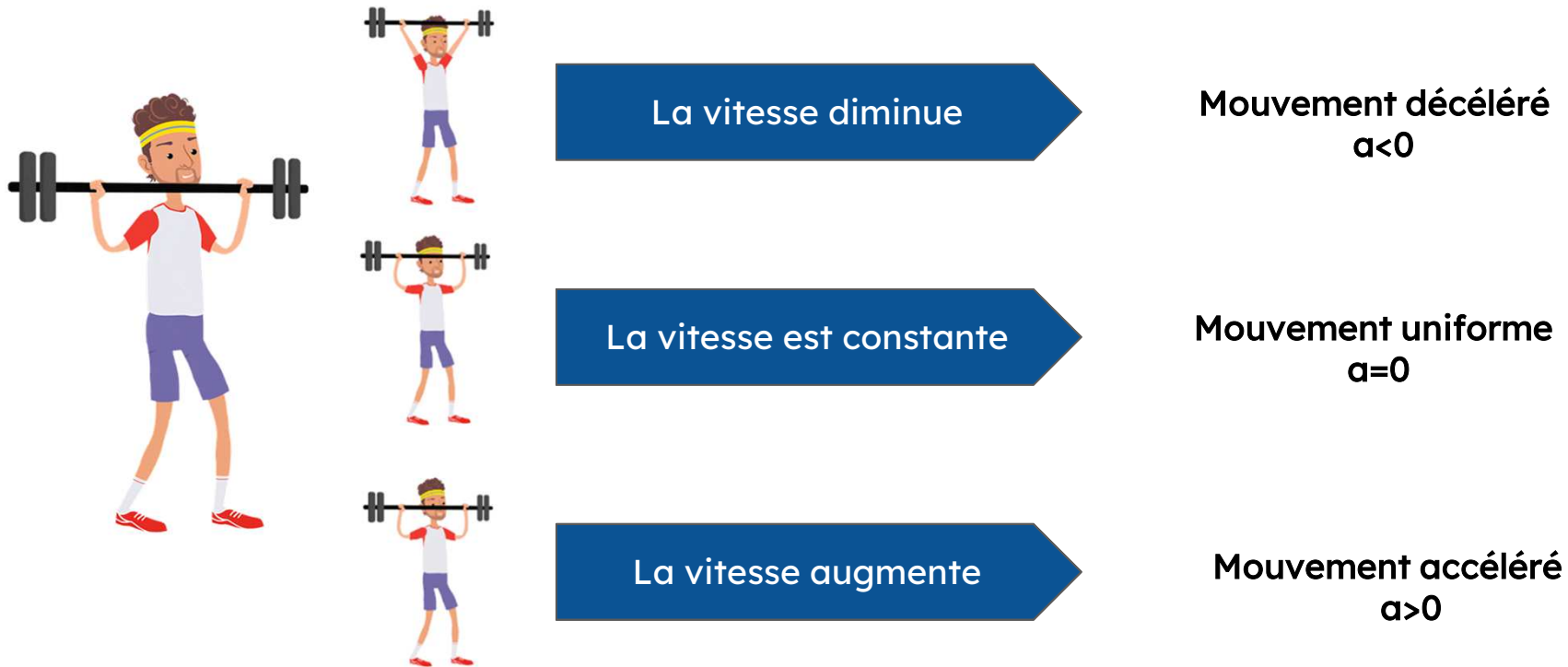
- Chaîne d'information.

## Concevoir, créer, réaliser

Compétences travaillées

Imaginer, concevoir et programmer des applications informatiques nomades.

# Comment peut-on comptabiliser les mouvements du sportif ?



# Comment détecter l'accélération de l'haltère en mouvement ?



Mouvement d'un smartphone ?

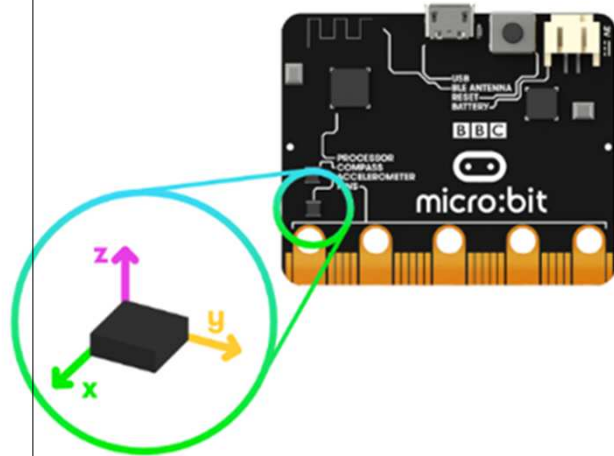


Accéléromètre (capteur d'accélération)



Exemples d'objets intégrant un accéléromètre

# Comment détecter les mouvements du sportif ?

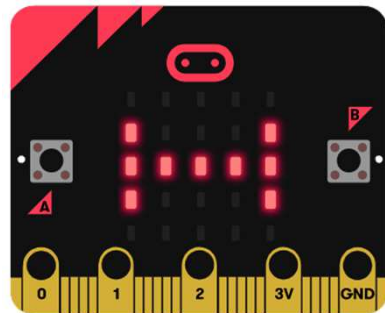


Capteur accéléromètre  
intégré dans la carte de  
programmation micro:bit



Carte micro:bit intégrée à un  
haltère

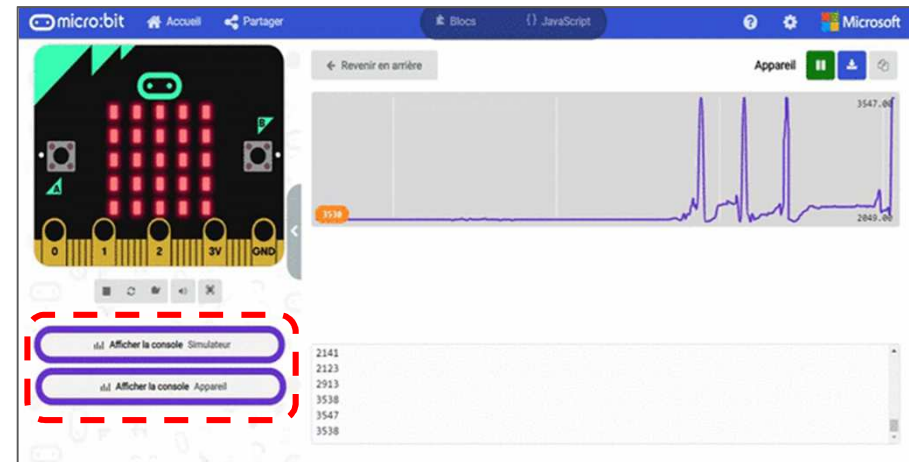
# Comment détecter les mouvements du sportif ?



<https://makecode.microbit.org>

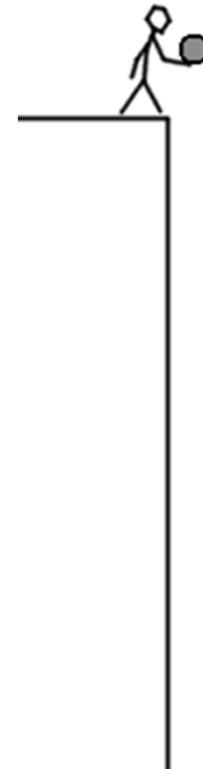
# Comment détecter les mouvements du sportif ?

## PROGRAMME PAR BLOCS



Nous allons fixer notre seuil de détection à 2500 milliG (mG)

# L'accélération ?





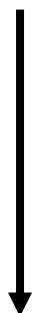
# Comment comptabiliser les mouvements du sportif ?



aller



retour



Comptage des mouvements



# Pourquoi utiliser une variable ?

Données météo :

Ensoleillé

Code postal :

97502

Température (°C) :

21

Mémoriser une donnée à l'intérieur  
de la mémoire d'un système

Mémoire du système

Variable

Les données  
stockées évoluent  
dans le temps

définir Lumière ▾ à niveau d'intensité lumineuse

Intensité lumineuse  
variable dans le temps

# Comment comptabiliser les mouvements du sportif ?

## ALGORITHME

### INITIALISATION

Afficher le logo d'un haltère  
Déclarer la variable "mvt" à 0

### PROGRAMME

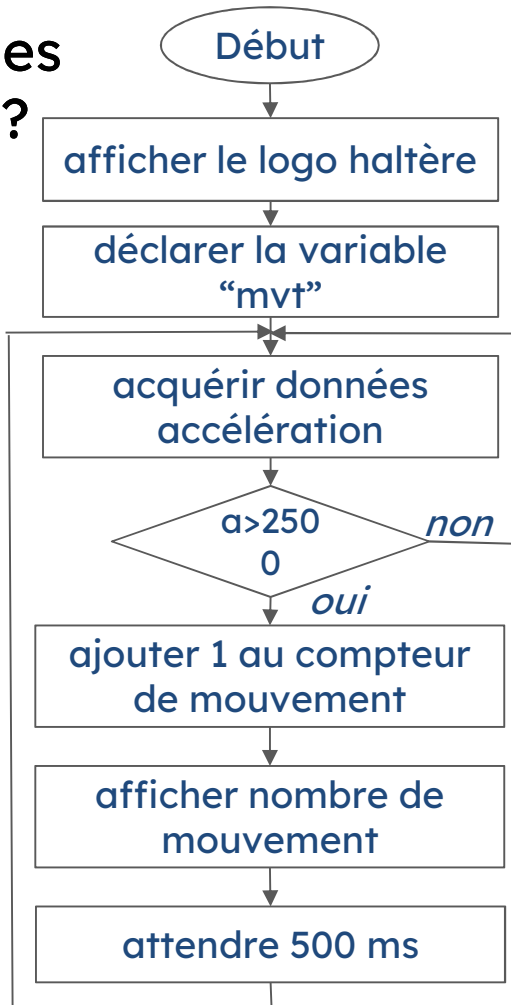
Il faut acquérir les données d'accélération

Si la valeur de l'accélération dans n'importe quelle direction est supérieur à 2500 mg

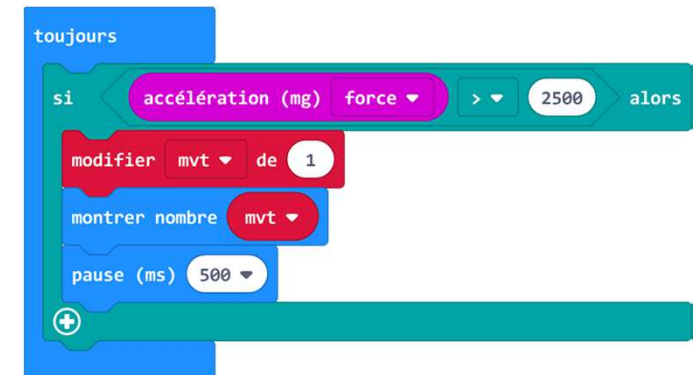
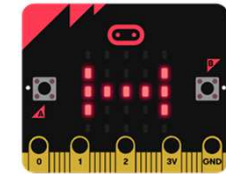
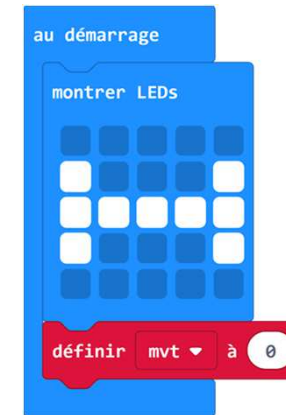
Alors :

Incrémenter la variable "mvt" de 1  
Afficher la variable "mvt" sur l'écran matriciel  
Et attendre 500ms

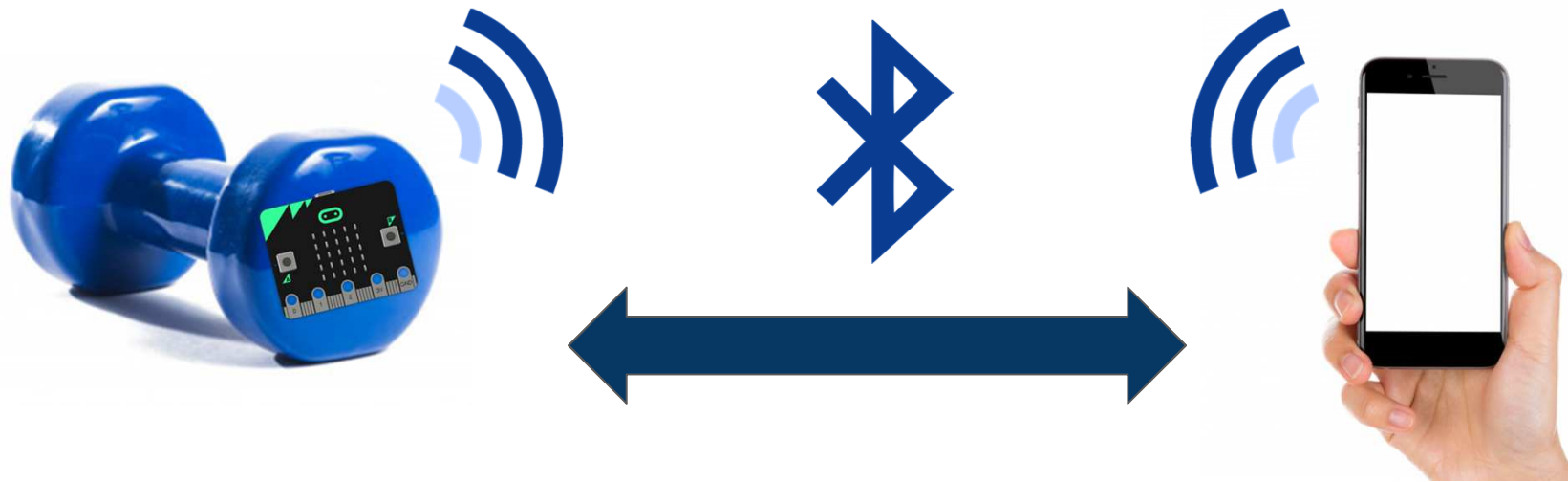
ALGORIGRAMME



## PROGRAMME PAR BLOCS



# Comment visualiser les performances ?



Carte micro:bit  
intégrée à un haltère

Communication bilatérale par  
bluetooth entre la carte micro:bit  
et le smartphone

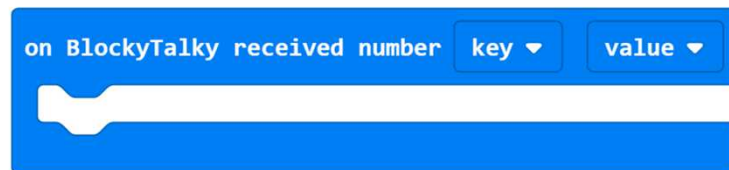
Visualiser ses  
performances

# Comment communiquer en Bluetooth avec la carte micro:bit ?



## PROTOCOLE DE COMMUNICATION

Pour communiquer des informations en Bluetooth, il faut signer ses données par une clé.  
*Exemple, transmission du nombre de mouvement en Bluetooth : geste24*  
*La clé est "geste". La valeur communiquée est 24 (mouvements)*



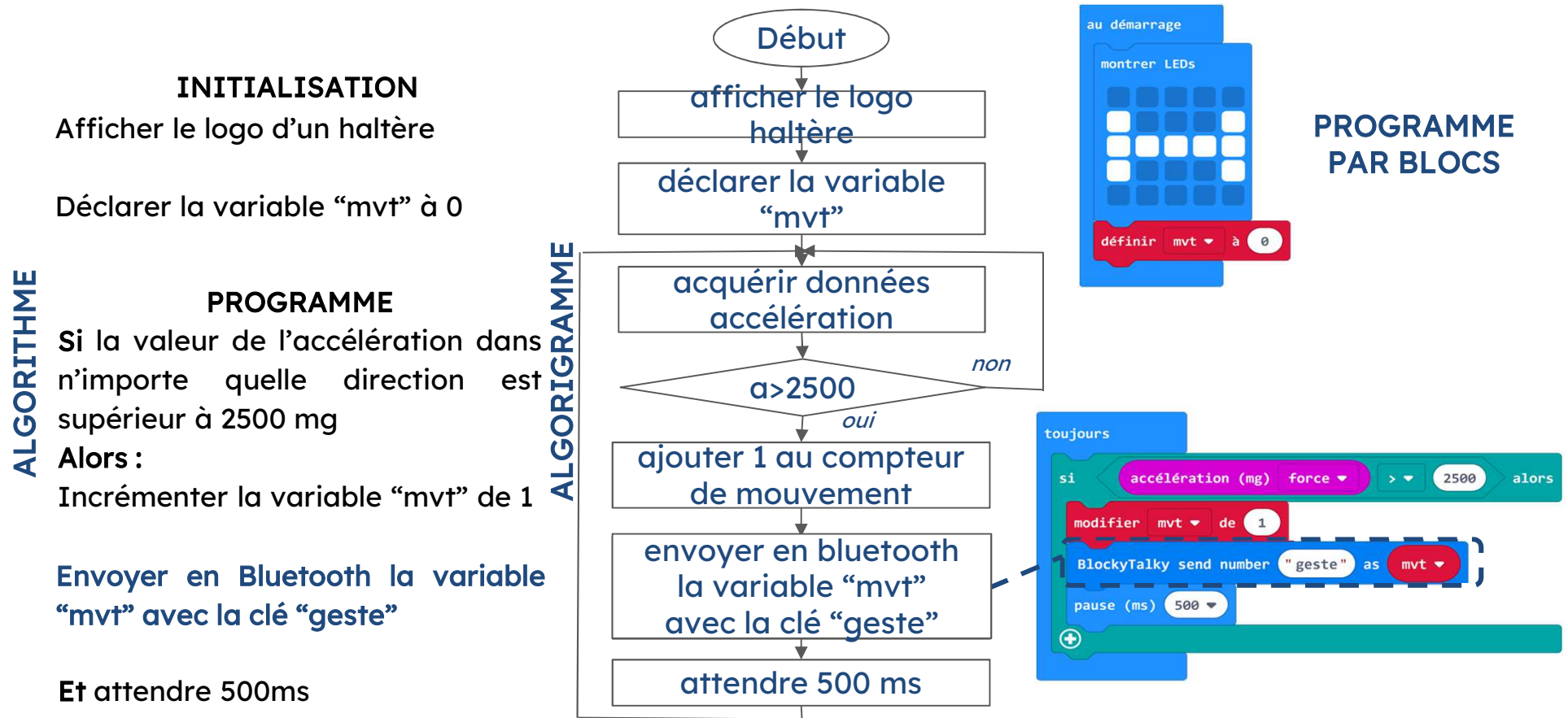
Permet d'envoyer des données par Bluetooth

- Choix de la clé à envoyer : geste
- Choix des données à envoyer : variable "mvt"

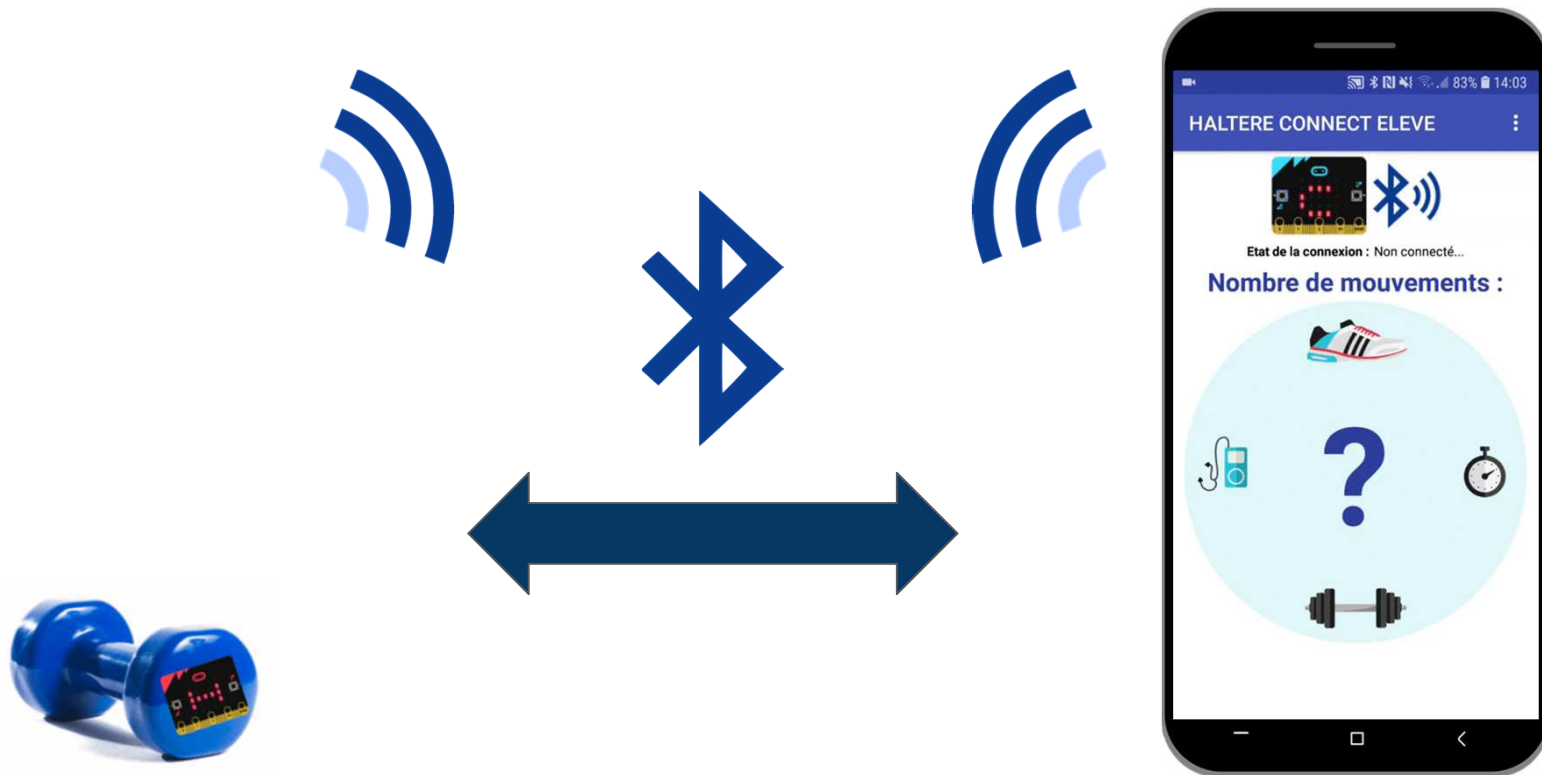
Permet de recevoir des données par Bluetooth

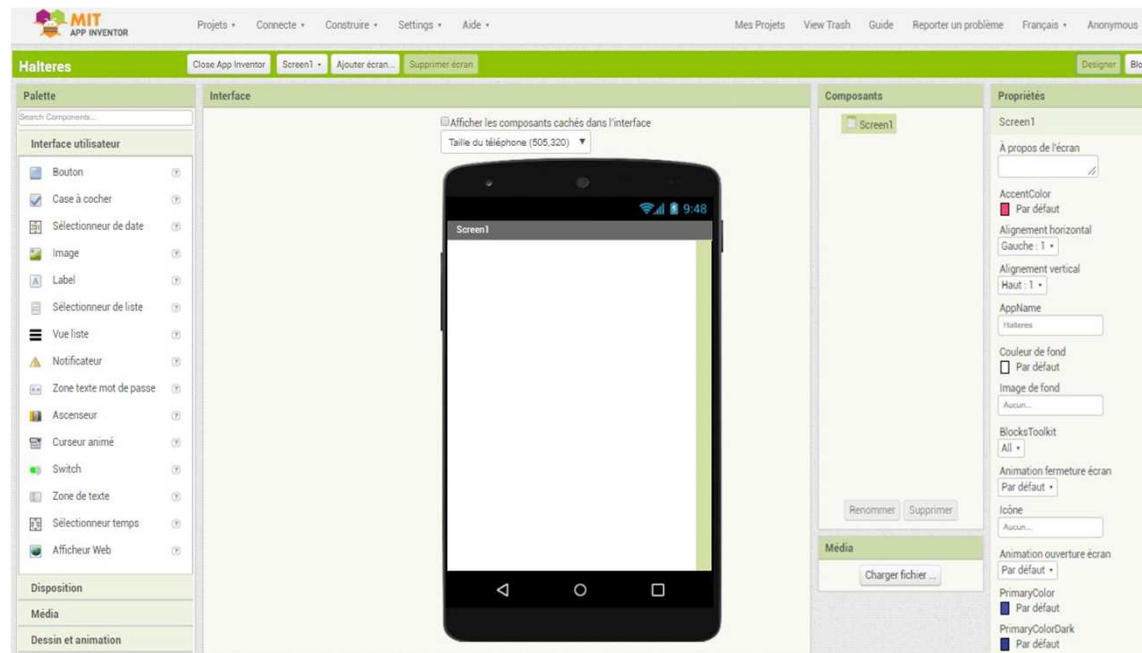
- Réception d'une clé (variable "key")
- Réception d'une donnée (variable "value")

# Comment communiquer en bluetooth avec la carte micro:bit ?



# Comment visualiser dans une application le nombre de mouvements ?





Outil numérique en ligne permettant de créer des applications Android



# Préambule

Pour simuler, il faut d'abord installer sur son smartphone l'application gratuite MIT AI2 Companion



MIT AI2 Companion

MIT Center for Mobile Learning Enseignement

★★★★★ 22033

1 PEGI 3

× Cette application n'est pas compatible avec votre appareil.

Installée

<https://frama.link/ai2>



*Lien pour télécharger l'application de base :*

<https://frama.link/haltere>

*Lien pour télécharger l'image du bouton init :*

<https://frama.link/bouton>

# Comment se connecter à l'outil numérique App Inventor ?



<http://code.appinventor.mit.edu>

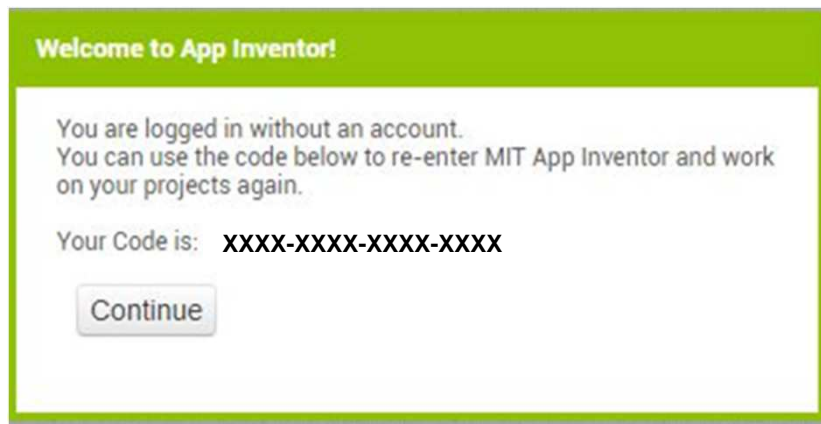


*Connexion sans  
création de compte*

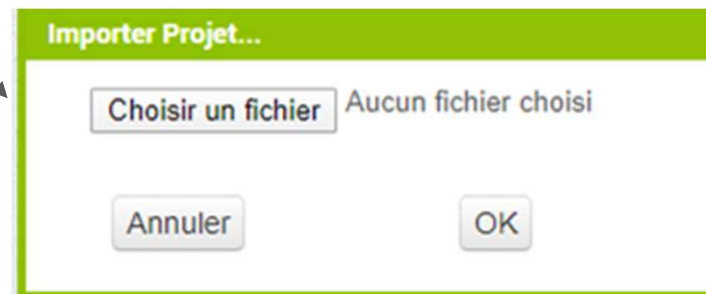
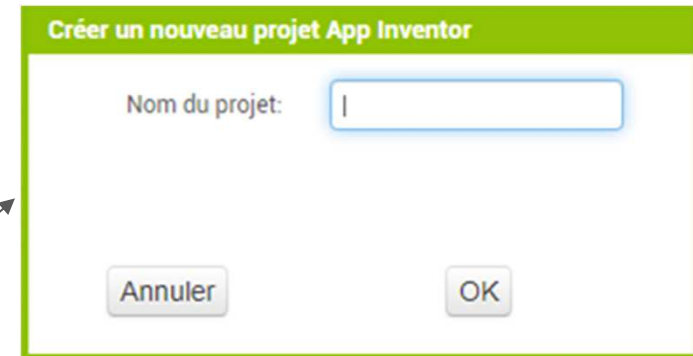
*Ne pas oublier de  
noter et conserver le  
code donné !!!*

# Première connexion à App Inventor

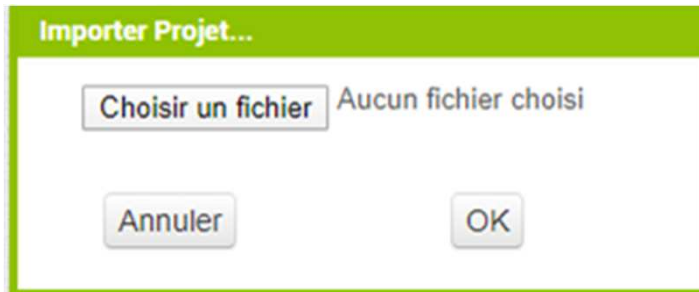
*Pensez à changer la langue*



*Code à noter et à conserver pour pouvoir accéder à ses fichiers*



# Comment importer l'application de base ?



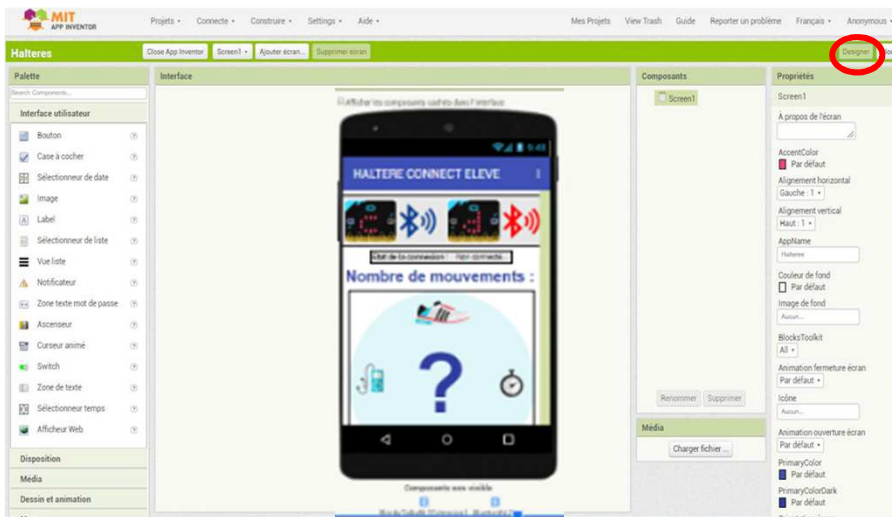
*Lien pour télécharger l'application de base :*

<https://frama.link/haltere>

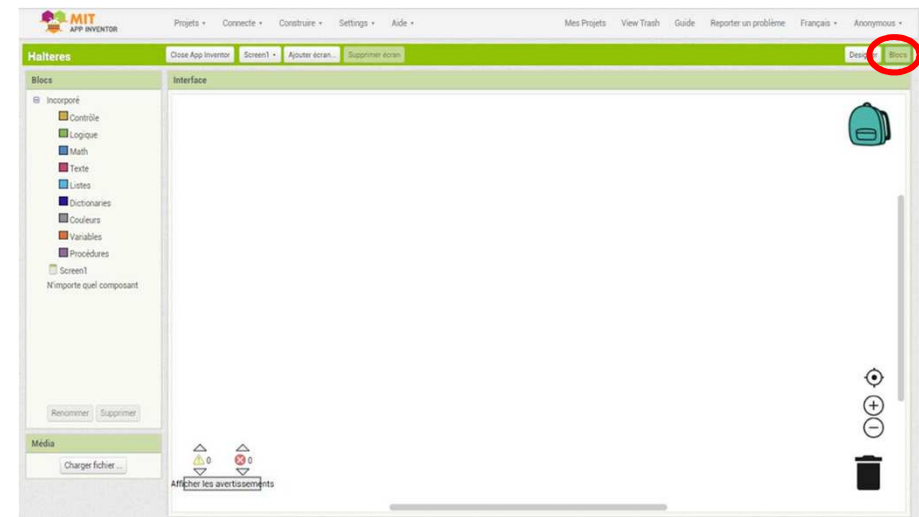


# Deux parties dans App Inventor

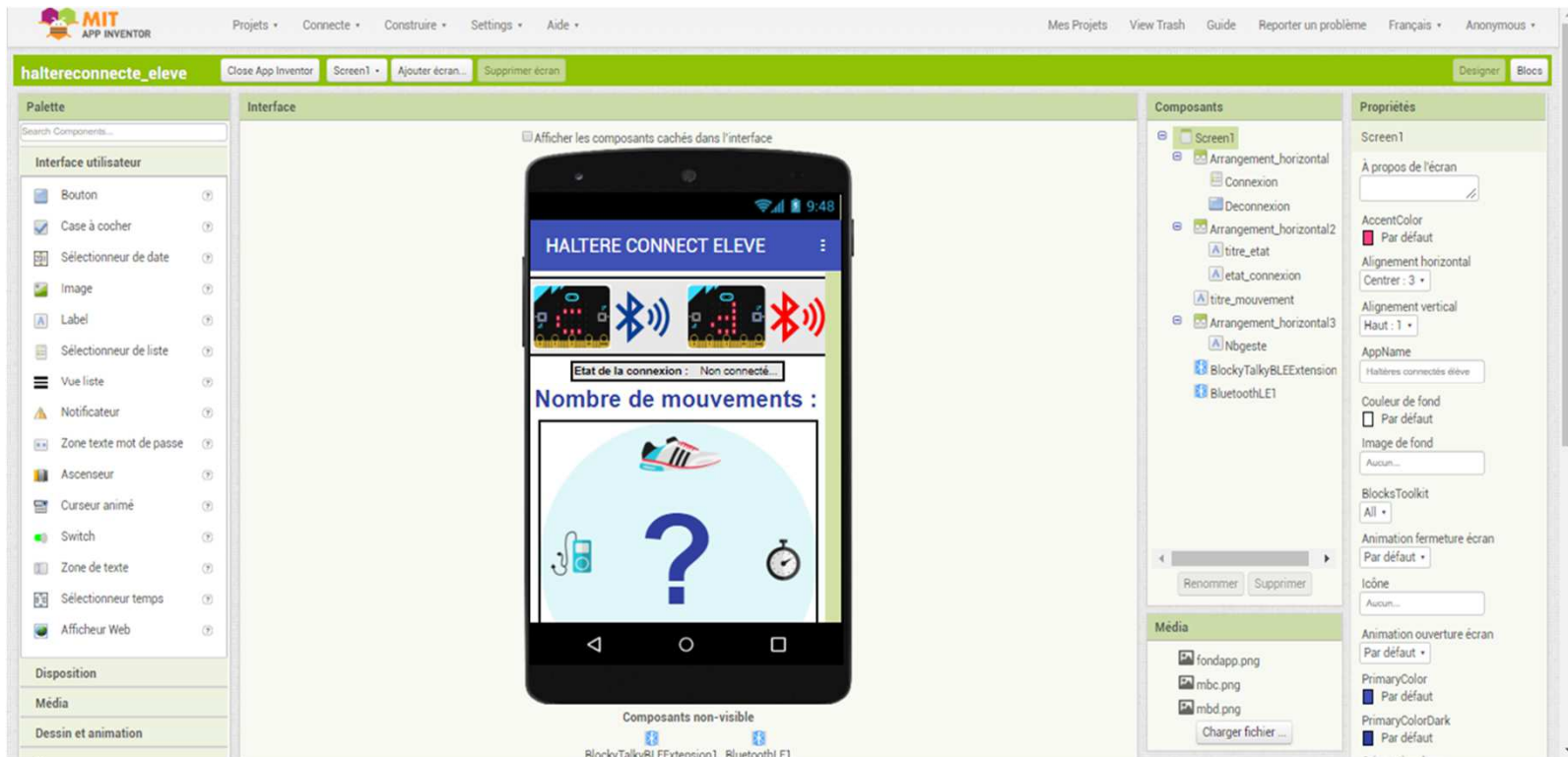
## Designer



## Blocs



# Partie designer de l'application de base



Palette

Interface

Composants

Propriétés

# Partie blocs de l'application de base

The screenshot displays the MIT App Inventor web interface for a project named "haltereconnecte\_eleve". The interface is divided into several sections:

- Top Bar:** Includes the MIT App Inventor logo, navigation menus (Projets, Connecte, Construire, Settings, Aide), and user information (Mes Projets, View Trash, Guide, Reporter un problème, Français, Anonymous).
- Project Bar:** Shows the project name "haltereconnecte\_eleve" and buttons for "Close App Inventor", "Screen1", "Ajouter écran...", and "Supprimer écran".
- Left Panel (Blocs):** A sidebar containing a "Blocs" section with categories like "Contrôle", "Logique", "Math", "Texte", "Listes", "Dictionnaires", "Couleurs", "Variables", and "Procédures". Below this is a "Média" section with files "mbc.png" and "mbd.png", and a "Charger fichier..." button.
- Main Canvas (Interface):** The central workspace where blocks are assembled. It shows:
  - Event blocks: "quand Screen1 .Initialise ...", "quand Connexion .Après pri...", "quand BluetoothLE1 .Device...", and "quand Deconnexion .Clic fa...".
  - Logic block: "initialise global mouvementvar à 0".
  - Procedure block: "quand BlocklyTalkyBLEExtension1 .IntReceived" containing:
    - Table block with columns "key" and "value".
    - Condition block: "si obtenir key = 'geste'".
    - Then block: "alors" containing:
      - Block: "mettre global mouvementvar à obtenir value".
      - Block: "mettre Nbgeste . Texte à obtenir global mouvementvar".
- Right Panel:** Contains a "Designer" button, a "Blocs" button, and a trash icon.
- Bottom:** A footer with the text "Politique de confidentialité et Conditions d'utilisation".

# Programme de base de l'application

## ALGORITHMME

### INITIALISATION

Gérer la connexion Bluetooth à l'aide de deux boutons

Initialiser la variable "mouvementvar" à zéro

### PROGRAMME

Si on reçoit par bluetooth la clé "geste"

Alors :

Récupérer la valeur associée à la clé "geste" dans la variable "mouvementvar"

Afficher la variable "mouvementvar" dans le label "Nbgeste"

## PROGRAMME PAR BLOCS

```
quand Screen1 .Initialise ...
quand Connexion .Après pri...
quand BluetoothLE1 .Device...
initialise global mouvementvar à 0

quand BlockyTalkyBLEExtension1 .IntReceived
  key value
  faire
    si obtenir key = "geste"
      alors
        mettre global mouvementvar à obtenir value
        mettre Nbgeste . Texte à obtenir global mouvementvar
```



# Comment réinitialiser le nombre de mouvement ?



# Comment réinitialiser le comptage des mouvements depuis l'application ?

## Cahier des charges :



- ajouter un bouton de réinitialisation dans l'application ;
- améliorer l'esthétique du bouton en ajoutant une image illustratrice de la fonction "réinitialiser" ;
- programmer le bouton pour afficher la valeur "0" dans l'application ;
- programmer le bouton pour envoyer par Bluetooth la valeur "0" signée avec la clé "init".

# Comment réinitialiser le comptage des mouvements depuis l'application ?

## ALGORITHMME

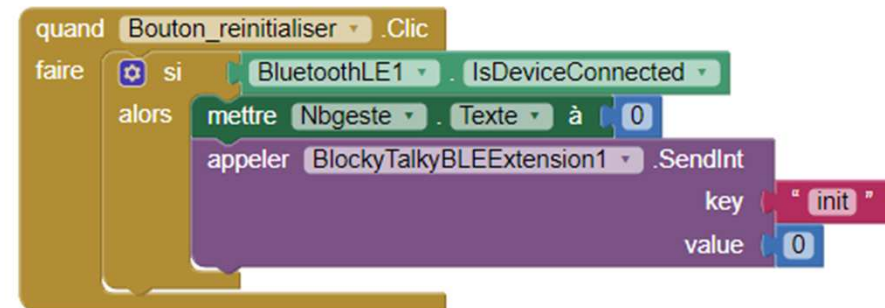
Si j'appuie sur le bouton 'réinitialiser' et Si la connexion Bluetooth est active

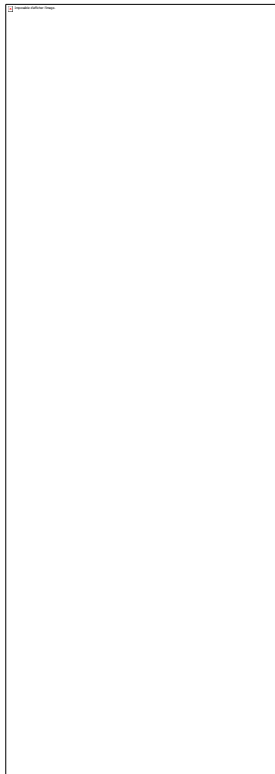
**Alors :**

Afficher la valeur "0" dans le label "Nbgeste"

Envoyer la valeur "0" associée à la clé 'init' par Bluetooth à la carte micro:bit

## PROGRAMME PAR BLOCS



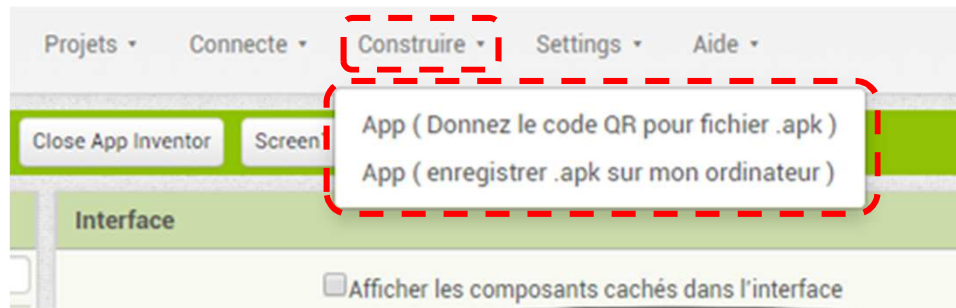


# Passons à la pratique

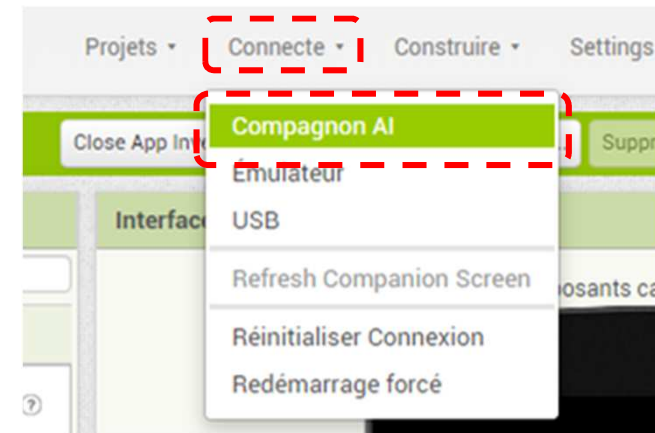


# Comment tester l'application ?

## Deux possibilités

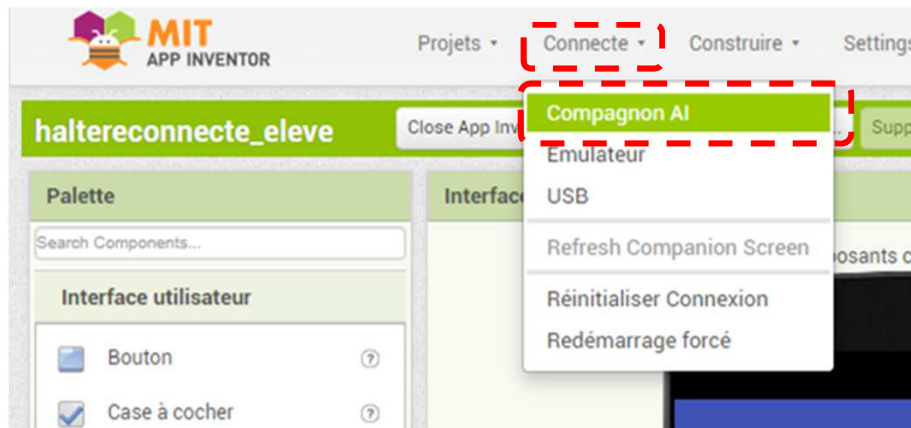


**Générer et installer  
l'application sur un  
smartphone**



**Simuler le  
fonctionnement à  
l'aide d'un smartphone**

# Comment simuler l'application sur un smartphone ?



Fonction “Compagnon AI” permettant de simuler

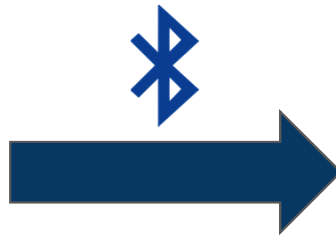


QRcode ou code à lire à partir de l'application MIT AI2 Companion installée sur le smartphone.

# Comment récupérer les données transmises par Bluetooth pour réinitialiser le nombre de mouvements ?



Bouton appuyé



Réinitialisation du nombre de mouvements

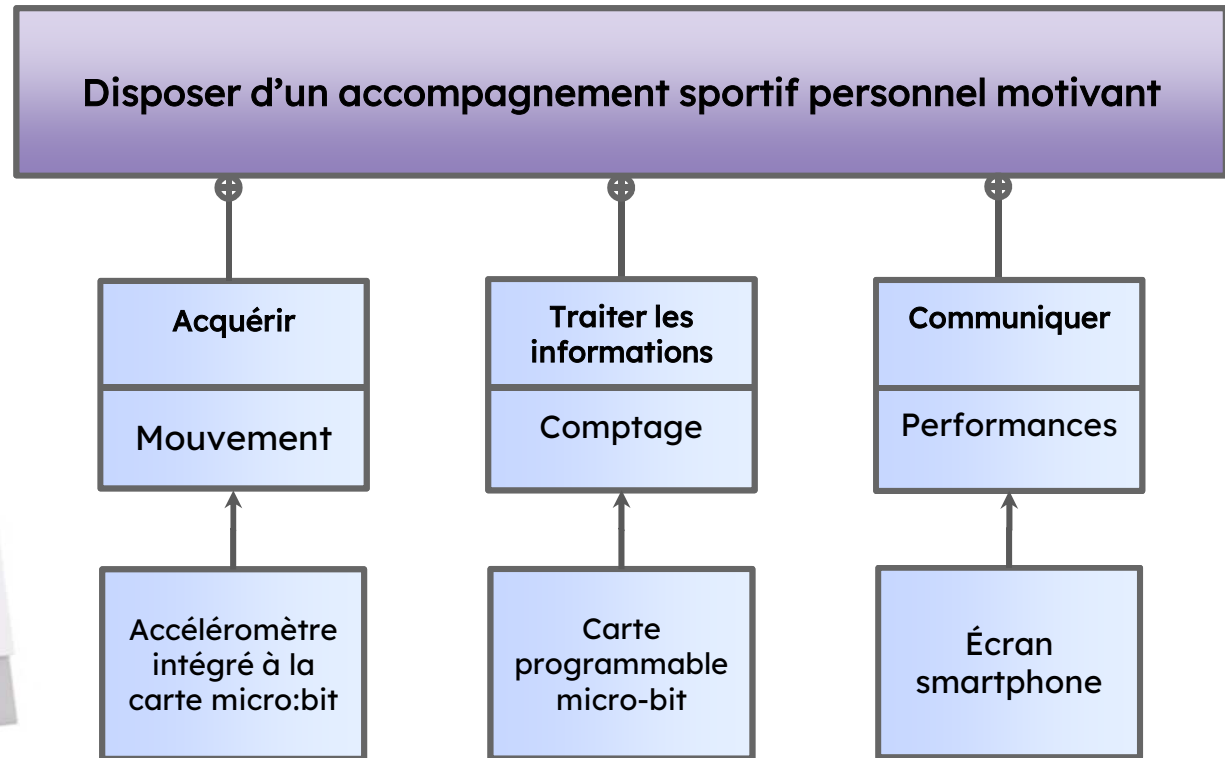
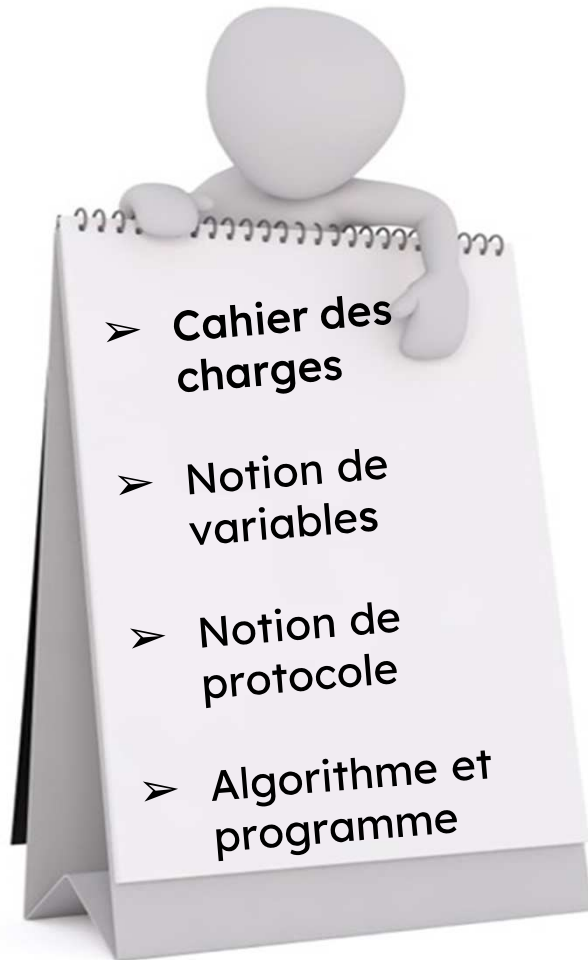
## ALGORITHMME

Si on reçoit la clé "geste" par Bluetooth  
**Alors :**  
Définir la variable "mvt" avec la valeur reçue  
*(la valeur reçue depuis l'application est 0)*

## PROGRAMME PAR BLOCS

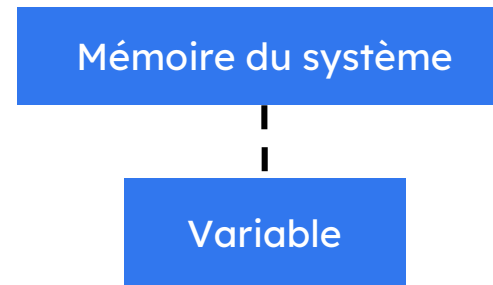
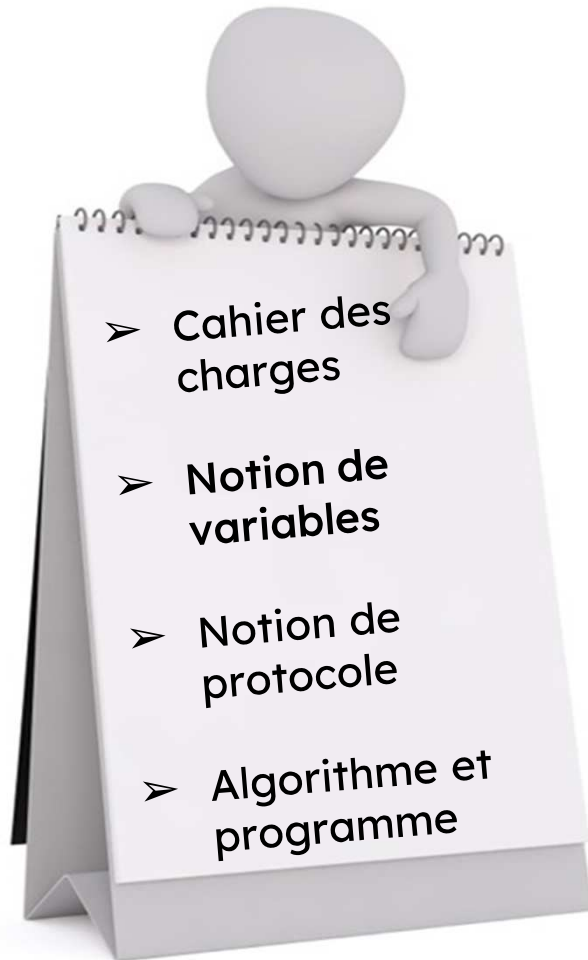
```
on BlocklyTalky received key and number value key value
si key = "init" alors
  définir mvt à value
  +
```

# Synthèse





# Synthèse




Les données stockées évoluent dans le temps

définir **Lumière** à **niveau d'intensité lumineuse**

Intensité lumineuse variable dans le temps

# Synthèse

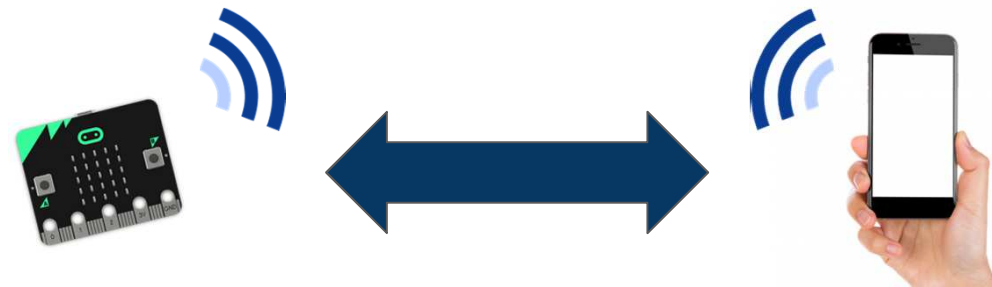
## PROTOCOLE DE COMMUNICATION ENTRE UNE CARTE MICRO:BIT ET APP INVENTOR

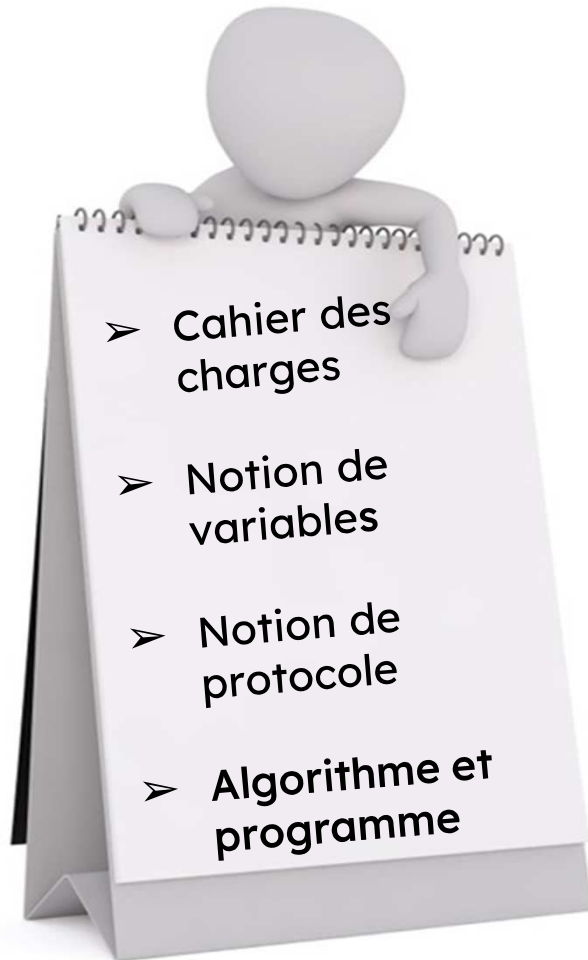
- 
- Cahier des charges
  - Notion de variables
  - Notion de protocole
  - Algorithme et programme

Pour communiquer des informations en Bluetooth, il faut **signer ses données par une clé.**

*Exemple, transmission de la température en Bluetooth :  
**temp24***

*La clé est "**temp**". La valeur communiquée est **24** (°C)*





# Synthèse

Un algorithme est un enchaînement de tâches ordonnées afin d'obtenir un résultat en utilisant un langage naturel et des mots clés : **si, alors, tant que, jusqu'à ...**

**Exemple d'algorithme permettant de compter les mouvements du sportif**

**Si** la valeur de l'accélération dans n'importe quelle direction est supérieur à 2500 mg

**Alors :**


Incrémenter la variable "mvt" de 1

Afficher la variable "mvt" sur l'écran matriciel

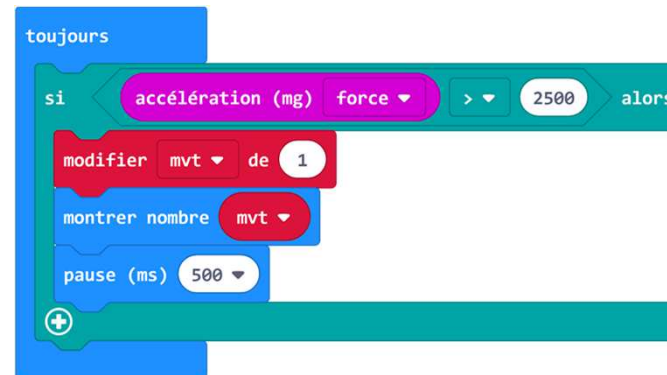
**Et attendre 500ms**

# Synthèse

Un programme par blocs est une représentation graphique d'un algorithme. Celui-ci sera compréhensible par un logiciel de programmation.

- 
- Cahier des charges
  - Notion de variables
  - Notion de protocole
  - Algorithme et programme

Exemple de programme par blocs permettant de compter les mouvements du sportif

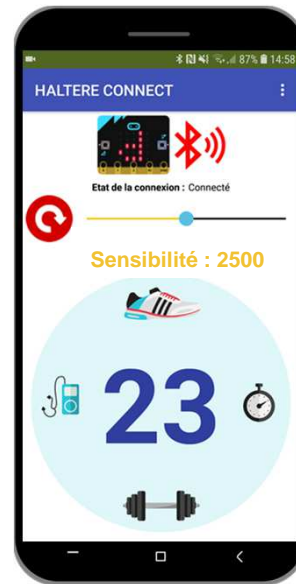


```
toujours
si <accélération (mg) force > > 2500 alors
  modifier mvt de 1
  montrer nombre mvt
  pause (ms) 500
+
```

# Piste de réflexion



Ajouter un chronomètre  
pour mesurer le temps  
de la séance

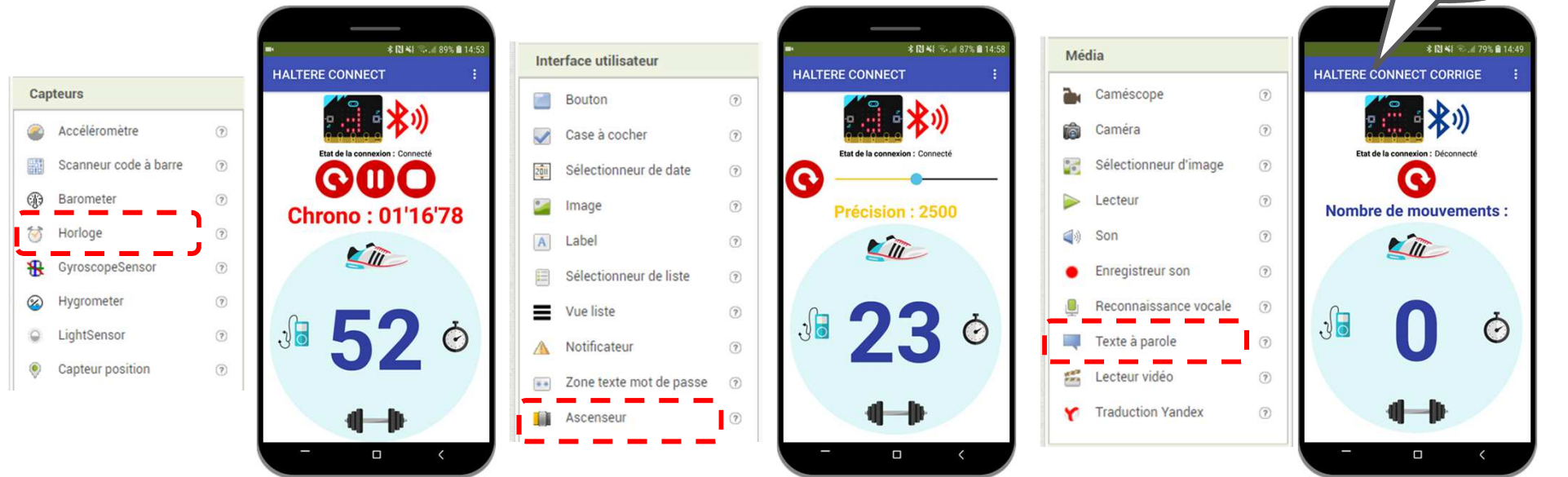


Ajouter un curseur pour  
pouvoir régler la sensibilité de  
la détection des mouvements



Faire parler  
l'application afin de  
motiver le sportif

# Piste de réflexion



Ajout de boutons et de labels et d'une fonction "Horloge"

Ajout d'un curseur (ascenseur)

Fonction "Texte à parole"

# Liens des programmes informatiques

Tracer les valeurs mesurées par l'accéléromètre : [https://makecode.microbit.org/\\_WFfD0gdDJc14](https://makecode.microbit.org/_WFfD0gdDJc14)

Comptabiliser les mouvements : [https://makecode.microbit.org/\\_d1AUoJMpAXCy](https://makecode.microbit.org/_d1AUoJMpAXCy)

Envoyer le nombre de mouvements par bluetooth : [https://makecode.microbit.org/\\_KvHe2xPe8haw](https://makecode.microbit.org/_KvHe2xPe8haw)

Application de base App Inventor (aia) : <https://frama.link/haltere>

Application corrigée App Inventor (aia) : <https://frama.link/halterecorrige>

Application de base à installer (apk) : <https://frama.link/haltereapk>

Application corrigée à installer (apk) : <https://frama.link/halterecorrigeapk>

Image du bouton init : <https://frama.link/bouton>

# Présentation de l'équipe

**Samuel VIOLLIN**

Inspecteur Général de l'éducation, du sport et de la recherche  
Doyen du groupe Sciences et Techniques Industrielles

**Thomas Roy**

Inspecteur d'Académie - Inspecteur Pédagogique Régional  
Sciences et Techniques Industrielles  
Corps d'inspection • Inspecteurs du second degré

**Rodolphe MOUIX**

Chargé de missions d'Inspection-professeur certifié de technologie

**Domenico LAZZARO**-professeur certifié de technologie

**Frédérique DEBEE**-professeur certifié de technologie

**Romain BERTRAND**-professeur certifié de technologie