Programmation du système BlazePod

Objectif pédagogique

Programmer un système interactif qui réagit à un mouvement lorsqu'il est activé par un interrupteur.

⚙️ **Matériel et entrées**

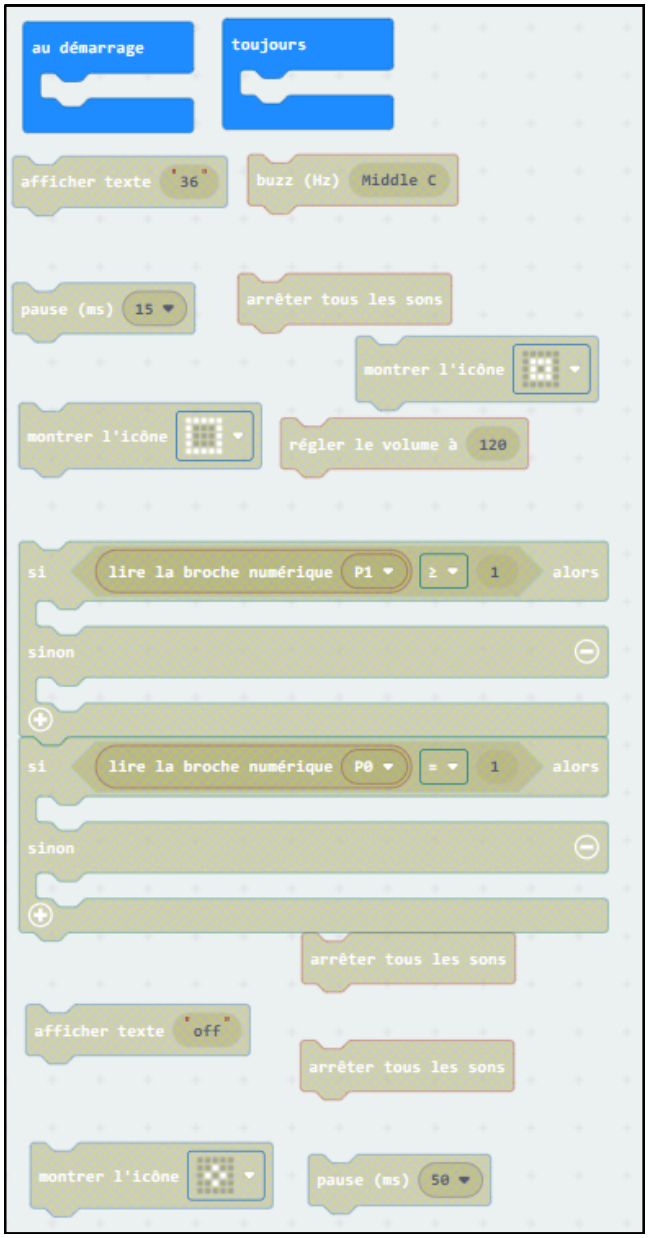
- P0 → Interrupteur (ON/OFF)
- P1 → Capteur de mouvement

🧩 **Consigne principale :** Créer un programme qui fonctionne comme un pod d'entraînement par détection de mouvement :

🔹 Fonctionnement attendu :

- Au démarrage :
  - Afficher un message (ex : "votre classe" ou un symbole)
  - Régler le volume sur 120
- En continu (boucle toujours) :
- Si interrupteur activé (P0 = 1) :
  - Si mouvement détecté (P1 = 1) :
    - Émettre un son
    - Afficher une icône (ex : cible touchée)
    - Faire une courte pause
    - Arrêter le son
    - Changer d'icône
  - Sinon :
    - Afficher une icône d'attente
    - Faire une petite pause
- Sinon (interrupteur OFF) :
  - Arrêter tous les sons
  - Afficher "OFF"

Liste des blocs nécessaires



✅ Critères de réussite

✓ Le système réagit uniquement si l'interrupteur est activé

✓ Le capteur déclenche une action visible + sonore

✓ Le programme fonctionne en boucle

✓ Le comportement est clair et lisible (icônes, sons)

Défi d'amélioration – Pod interactif (partie 1)

Ton pod fonctionne ? Bravo 👏

Tu dois maintenant améliorer ton programme en ajoutant une nouvelle fonctionnalité.

👉 **Consigne** :

Choisis UNE amélioration ci-dessous et implémente - la dans ton programme.

🔹 Améliorations faciles (recommandées)

🧮 **1. Compteur de points**

👉 But : compter le nombre de fois où le capteur détecte un mouvement

✔ Étapes :

Créer une variable appelée score

Mettre score = 0 au démarrage

Quand un mouvement est détecté :

ajouter 1 à score

Afficher le score avec "afficher nombre"

👉 Blocs à utiliser :
Variables → créer / modifier
Math → ajouter 1
Base → afficher nombre

⏱️ **2. Temps de réaction**

👉 But : mesurer la rapidité de l'utilisateur

✔ Étapes :

Quand le pod s'active → noter le temps

Quand le mouvement est détecté :

calculer le temps écoulé

afficher ce temps

👉 Indice :
Utilise le bloc "temps de fonctionnement (ms)"

👉 Blocs à utiliser :

Variables
Entrées / logique
Temps de fonctionnement

🔊 **3. Sons différents**

👉 But : améliorer le feedback sonore

✔ Étapes :

Choisir :

un son quand on réussit

un son différent quand on ne touche pas

👉 Blocs à utiliser :

Musique → jouer une note / mélodie

Défi d'amélioration – Pod interactif (partie2)

💡 4. Animation LED

👉 But : rendre le pod plus visuel

✓ Étapes :

Faire clignoter une icône avant l'activation

OU

Créer une animation quand le mouvement est détecté

👉 Blocs à utiliser :

Base → montrer l'icône
Base → pause

🔁 5. Mode ON/OFF amélioré

👉 But : rendre le système plus clair

✓ Étapes :

Si OFF → afficher une icône spécifique

Au démarrage → afficher "READY"

👉 Blocs à utiliser :

Base → afficher texte
Conditions (si / sinon)
🔹 Améliorations un peu plus difficiles

🎯 6. Objectif à atteindre

👉 But : créer un mini-jeu

✓ Étapes :

Fixer un objectif (ex : 10 points)

Si score atteint :

afficher "GAGNÉ"

jouer un son

⏳ 7. Temps limité

👉 But : jouer contre la montre

✓ Étapes :

Lancer un chronomètre (ex : 10 secondes)

Compter les points pendant ce temps

À la fin → afficher le score

🚦 8. Temps aléatoire

👉 But : rendre le jeu imprévisible

✓ Étapes :
Ajouter une pause aléatoire avant activation

👉 Indice :
Bloc "nombre aléatoire entre ... et ..."

✅ Critères de réussite 

✓ L'amélioration fonctionne correctement
✓ Le programme reste lisible
✓ Tu es capable d'expliquer ton code