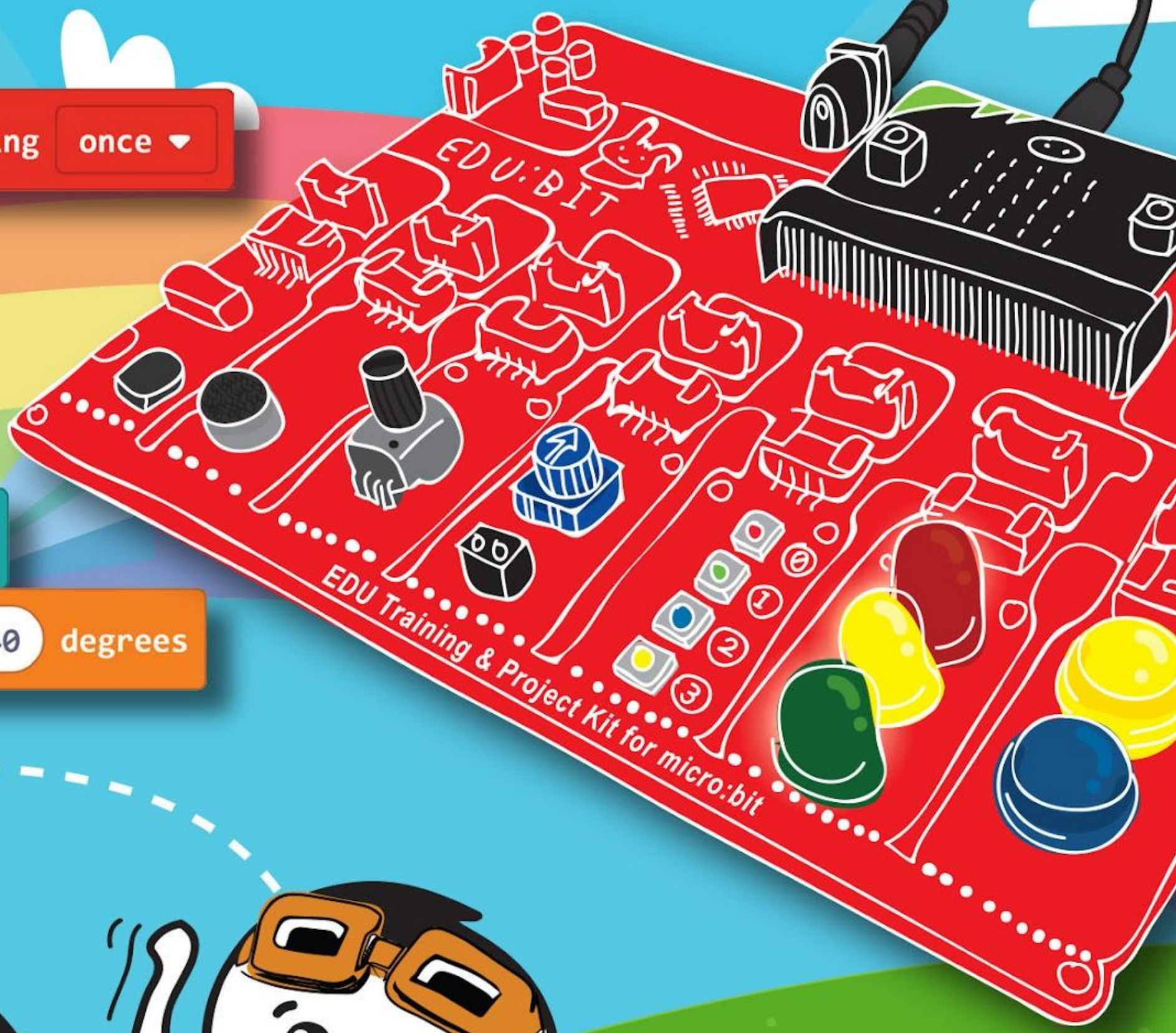


Explore STEM & Coding with

EDU:BIT

```
on start
  start melody power up repeating once
  set all RGB pixels to [green]

forever
  if IR sensor triggered then
    Set servo S1 position to 40 degrees
    show leds
  else
    Set servo S1 position to 20 degrees
    show arrow East
```



NOTE DE RERO EDUTEAM @ CYTRON

Cher / Chère _____ ,
(nom de l'enfant)

As-tu déjà entendu parler du micro:bit ? Il s'agit d'une petite carte programmable créée au Royaume-Uni et distribuée dans le monde entier pour encourager les enfants à apprendre à coder, simplement et ludiquement.

Les ingénieurs de Cytron sont allés encore plus loin en créant la carte EDU:BIT afin que tu puisses apprendre à coder pas-à-pas. Tu y trouveras différents modules : **Music Bit** muni d'un buzzer piézoélectrique et une prise jack pour écouter de la musique, **Sound Bit** pour détecter du bruit, **Potentio Bit** pour un contrôle analogique, **IR Bit** pour détecter des objets, **RGB Bit** pour allumer des lumières de toutes les couleurs, **Traffic Light Bit** avec ses LEDs rouge, jaune et verte, et enfin **Button Bit**, des versions plus grandes et accessibles des boutons présents sur la carte micro:bit elle-même. Dans ce kit, tu trouveras également un **Moteur DC** et un **Servomoteur** pour t'amuser avec. Cool, n'est-ce pas ?!

Nous pouvons commencer. On y va ? Dans les pages suivantes, nous allons explorer et recréer quelques classiques des jeux pour enfants comme *Pierre - Papier - Ciseaux*, *Échelles et serpents*, *C'est toi le chat !*, *Incroyable talent*, *Twister*, *Jacques a dit* et d'autres jeux amusants. Suis le guide pas-à-pas pour créer ces jeux et t'amuser en jouant avec tes amis ! N'hésite pas à modifier les codes par la suite pour les améliorer, et proposer tes propres versions des différents jeux.

À la fin de chaque chapitre, on te propose un défi pour mettre en œuvre ce que tu as appris et créer une application pour ta salle de classe. Essaie, et sache que si tu es bloqué(e), nous serons toujours là pour t'aider.

Es-tu prêt(e) ? Embarquons pour un voyage excitant : découvrir et apprendre en s'amusant !

Bien sincèrement,
Adam & Anna



Exploration STEM et Codage avec EDU:BIT Kit Projets et Formation

Écrit par Cheryl Ng, SC Lim et Adrian Teo

Traduit par François-Xavier Déchamps

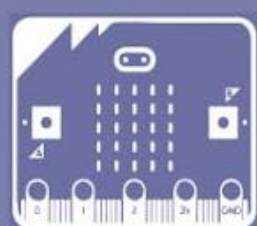
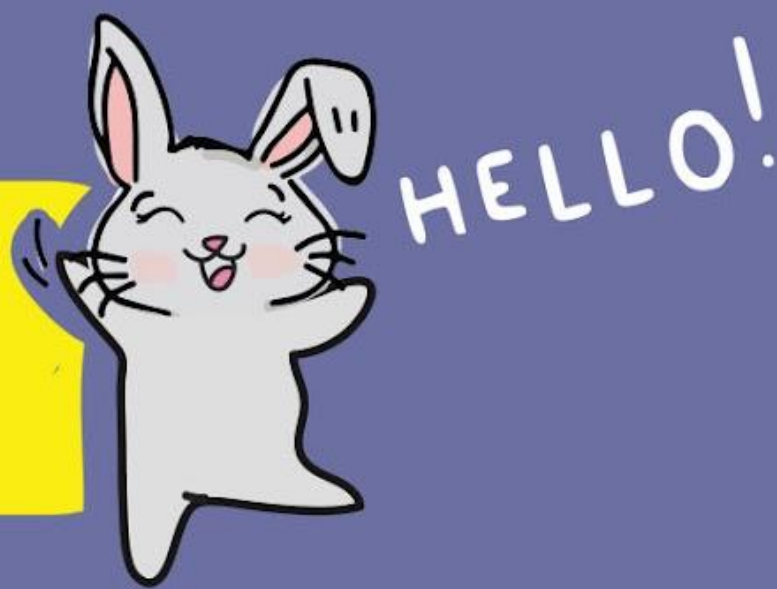
Illustré par Suhana Oazmi

2020

Publié par



Sommaire



Chapitre 1 : Bonjour, monde ! (Matrice de LEDs sur le micro:bit)
au démarrage et toujours

1 - 11



Chapitre 2 : Jouons à Pierre, Papier, Ciseaux ! (Button Bit)
Variables et programmation événementielle

12 - 25



Chapitre 3 : Un peu de Musique~ (Music Bit)
Fonctions en programmation

26 - 38



Chapitre 4 : Dessinez, c'est Gagné~ (Traffic Light Bit)
Sorties numériques

39 - 47



Chapitre 5 : Dé Numérique... Roule~ (IR Bit)
Entrées numériques, tableaux et boucles tant que

48 - 60



Chapitre 6 : C'est Toi le Chat ! (Potentio Bit)
Entrées analogiques, instructions conditionnelles

61 - 74



Chapitre 7 : Écoutons les Applaudissements ! (Sound Bit)
Changer de modes dans un programme

75 - 87



Chapitre 8 : Allons Faire un Tour ! (Moteur DC)
Contrôler le sens et la vitesse de rotation d'un moteur DC

88 - 95



Chapitre 9 : Tir au But... Goal ! (Servomoteur)
Contrôler la position du servomoteur

96 - 104



Chapitre 10 : Mastermind, Peux-Tu Déchiffrer le Code ? (RGB Bit)
Modèles de couleurs RVB

105 - 113



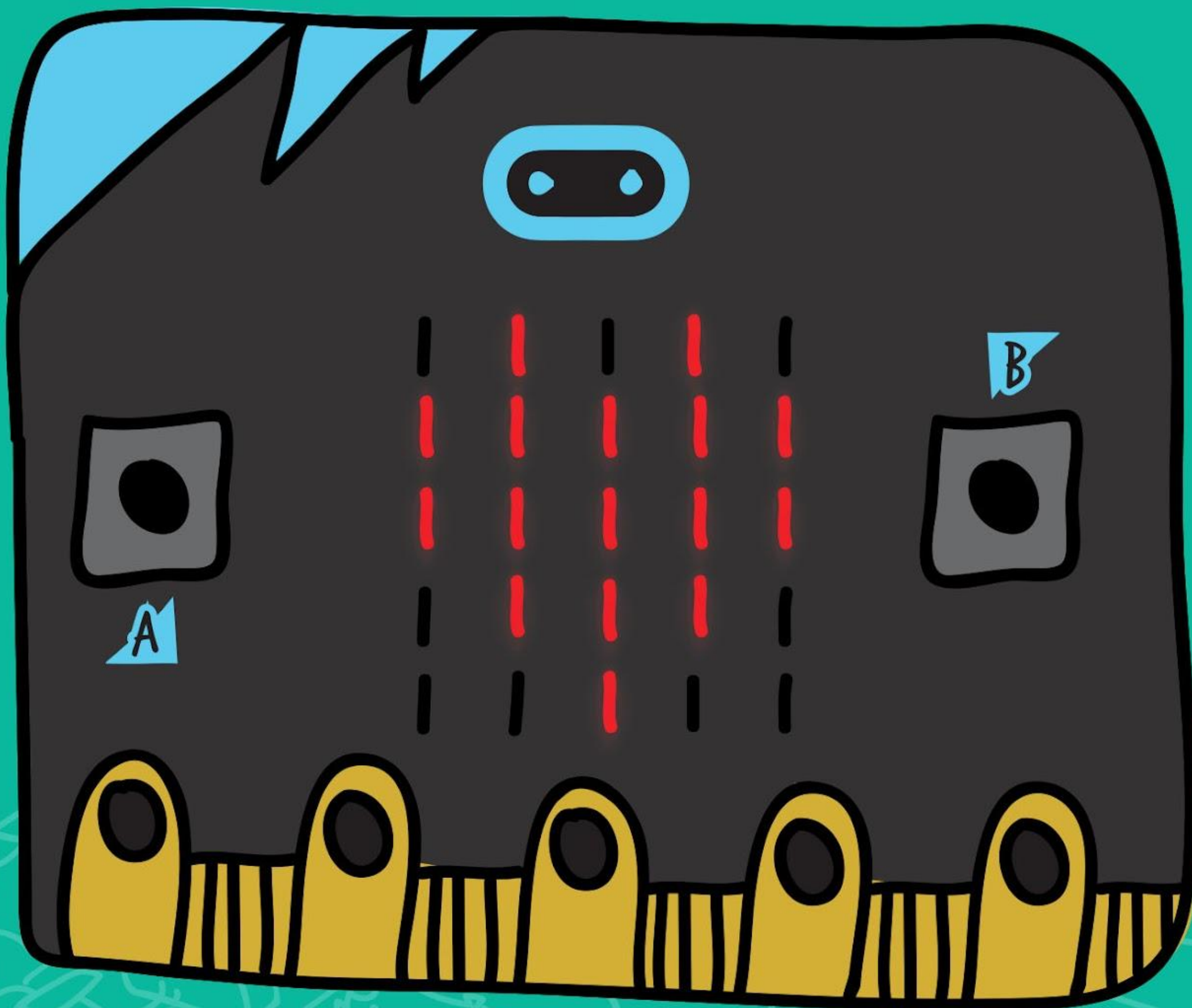
Chapitre Bonus : Jacques a dit, avec LEDs
Communication radio

114 - 124

CHAPITRE 1

> Bonjour, monde !

Matrice de LEDs sur le micro:bit

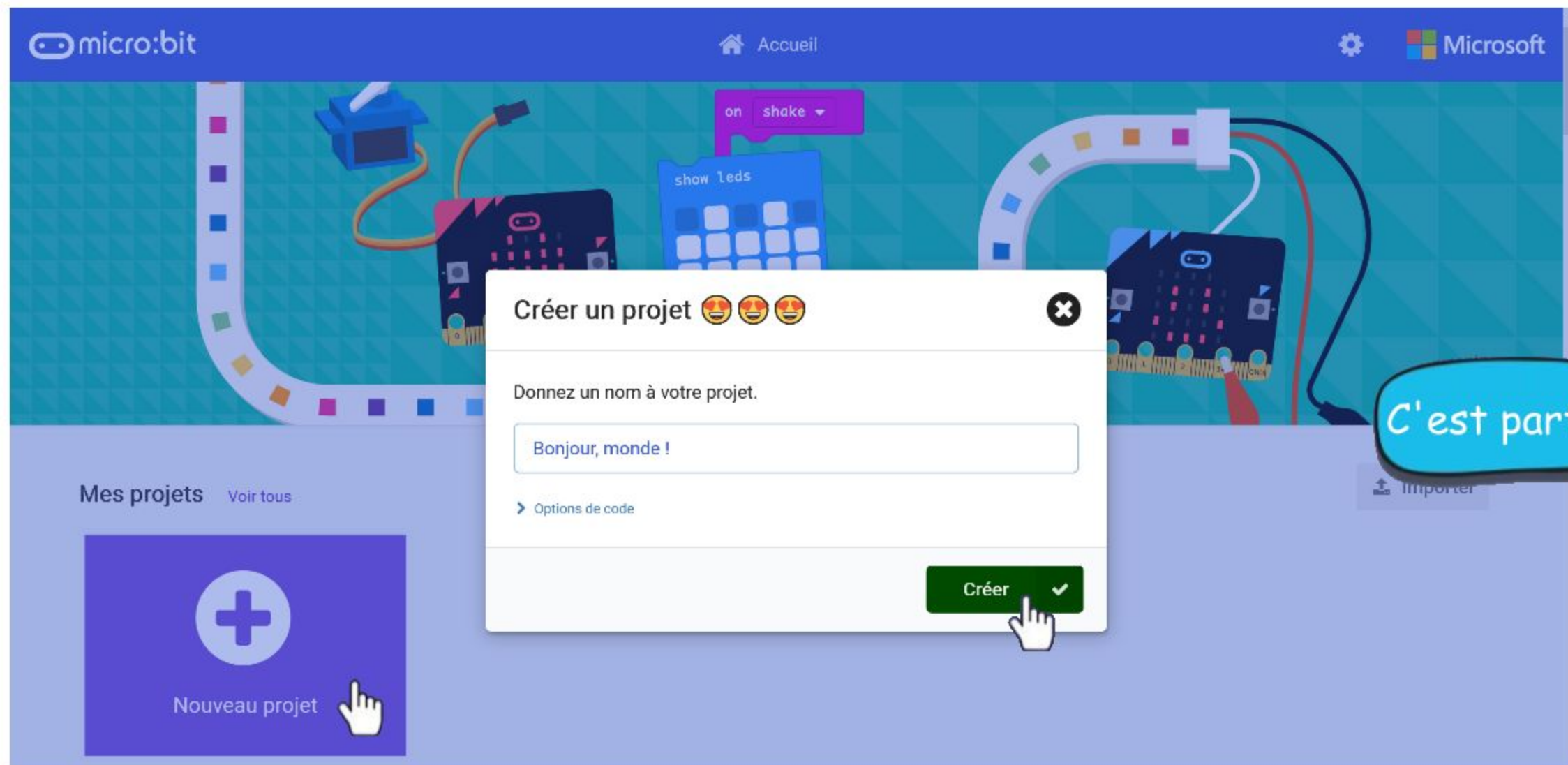


Scanne-moi !

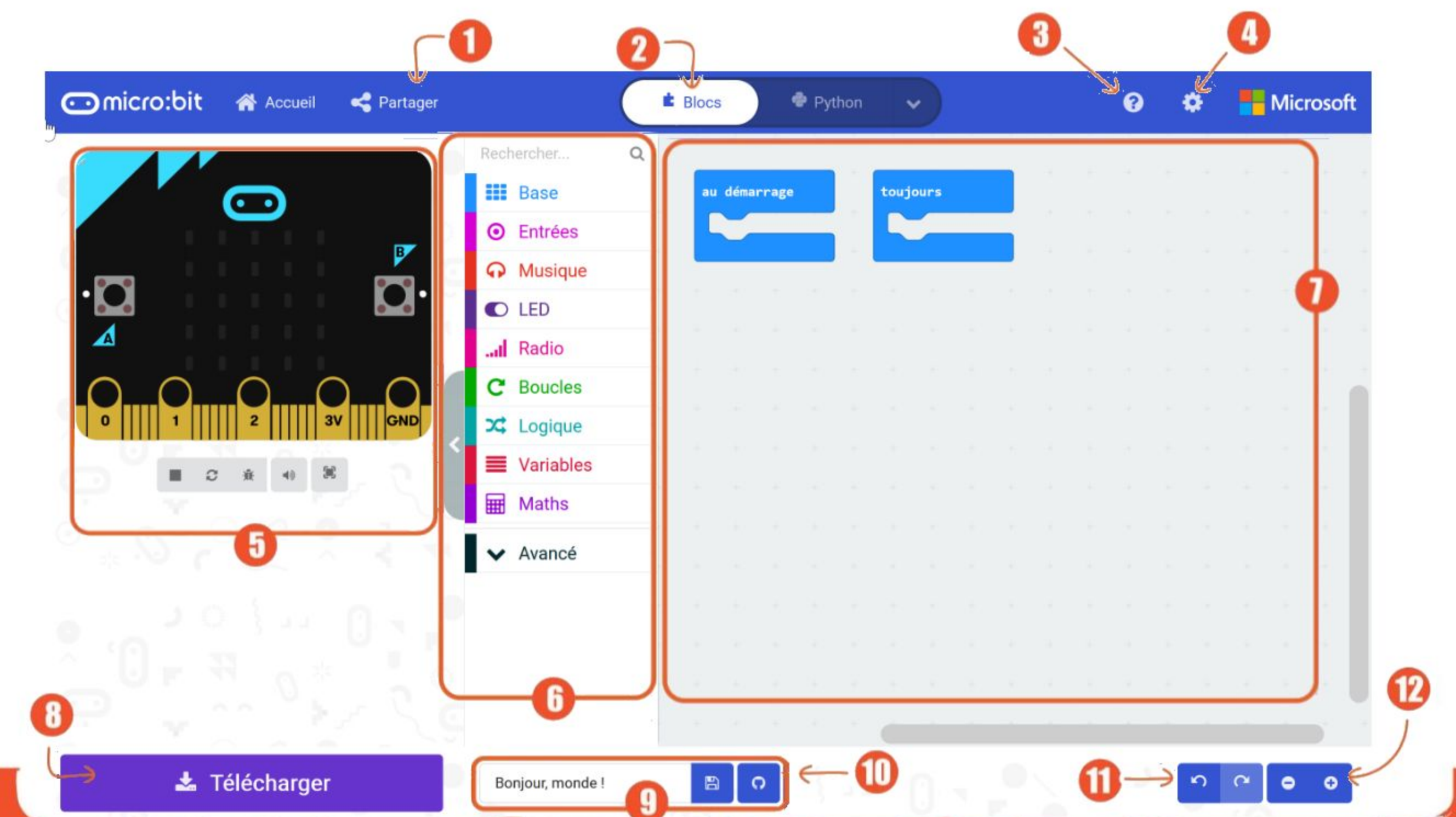
link.cytron.io/edubit-chapter-1

Codons !

Étape 1 Ouvre ton navigateur internet et vas sur <https://makecode.microbit.org/>
Clique sur **“Nouveau projet”**. Saisis le nom de ton projet puis clique sur **“Créer”**.



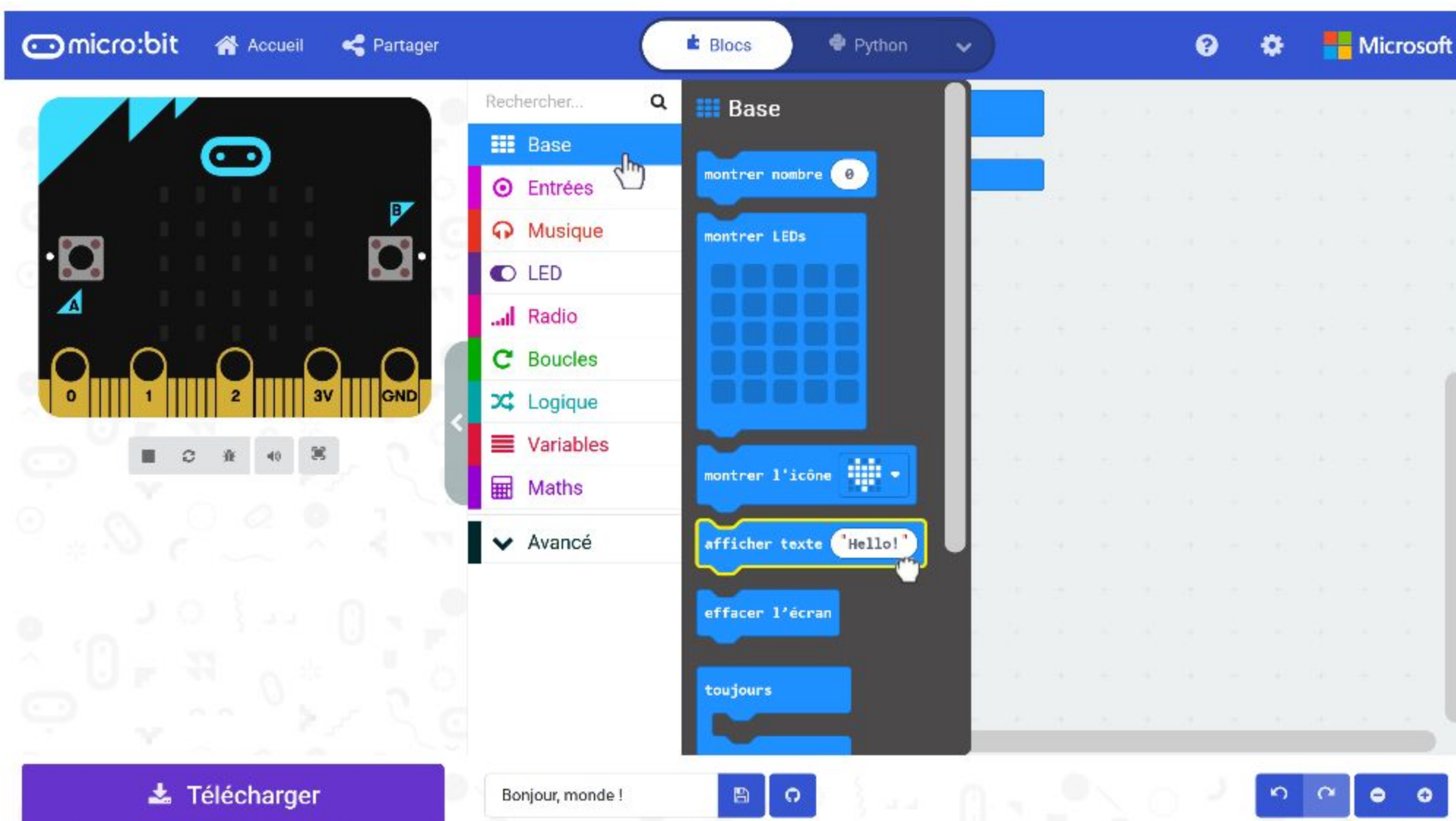
Tu peux voir l'éditeur de programme du logiciel Microsoft MakeCode qui te permettra de programmer ta carte EDU:BIT par la méthode du glisser-déposer.



- 1) Publie et partage ton projet
- 2) Choisis ton langage de programmation : Blocs, Javascript ou Python.
- 3) Ouvre le menu d'aide.
- 4) Change les paramètres, ajoute des extensions et appaie un périphérique.
- 5) **SIMULATEUR** - Affiche ce que ferait ton programme sur une carte micro:bit.
- 6) **BOÎTE À OUTILS / LISTE DES CATÉGORIES** - Clique pour voir les différents blocs disponibles dans chaque catégorie. Tous les blocs d'une même catégorie sont de la même couleur.
- 7) **ESPACE DE TRAVAIL** - Zone de création des programmes, qui sont faits de blocs qui s'imbriquent les uns avec les autres.
- 8) Télécharge ton code sur la carte micro:bit.
- 9) Nomme et enregistre ton projet en cours sur ton ordinateur.
- 10) Crée un dépôt GitHub.
- 11) Annuler / Rétablir.
- 12) Zoom arrière / avant.

CHAPITRE 1 : Bonjour, monde !

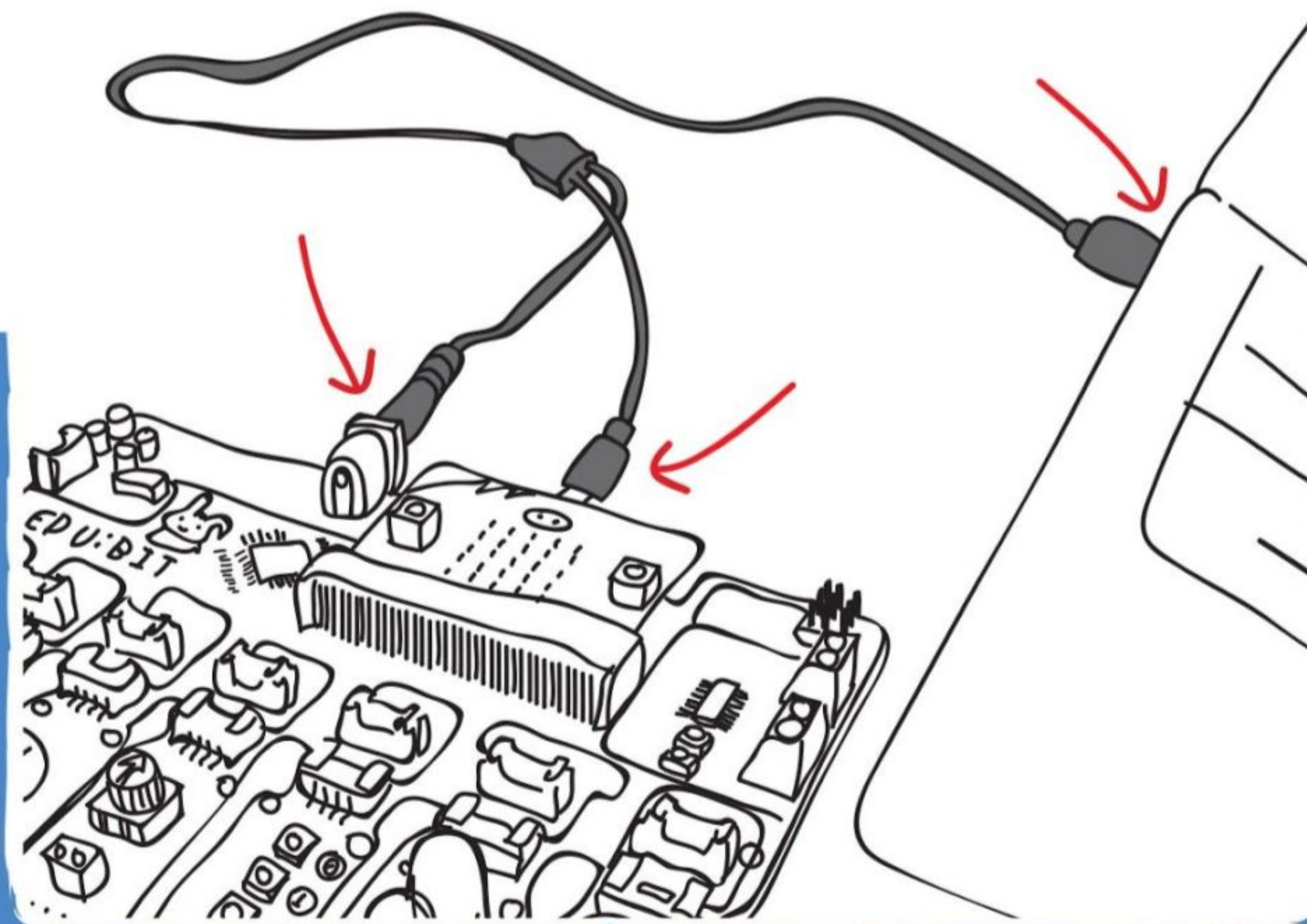
Étape 2 Clique sur [Base], puis clique et maintiens le clic sur le bloc [afficher texte].



Étape 3 Imbrique le bloc [afficher texte] dans le bloc [au démarrage].

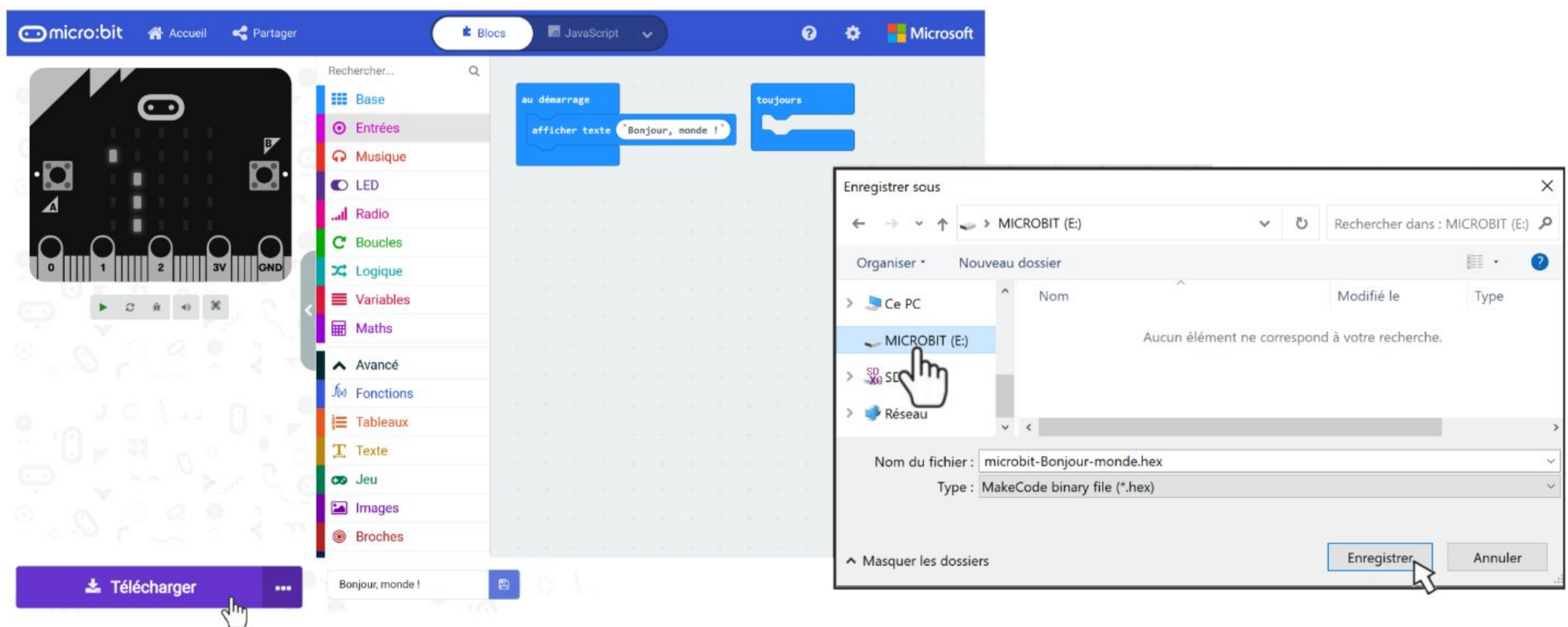


Étape 4 Connecte le câble USB sur ton ordinateur et sur EDU:BIT comme ci-dessous. Pense à allumer EDU:BIT en déplaçant l'interrupteur sur ON.





Étape 5 Clique sur le bouton [**Télécharger**]. Dans la fenêtre pop-up, sélectionne le lecteur MICROBIT, puis clique sur “Enregistrer”. Ferme la fenêtre lorsque le message “Téléchargement terminé...” apparaît.

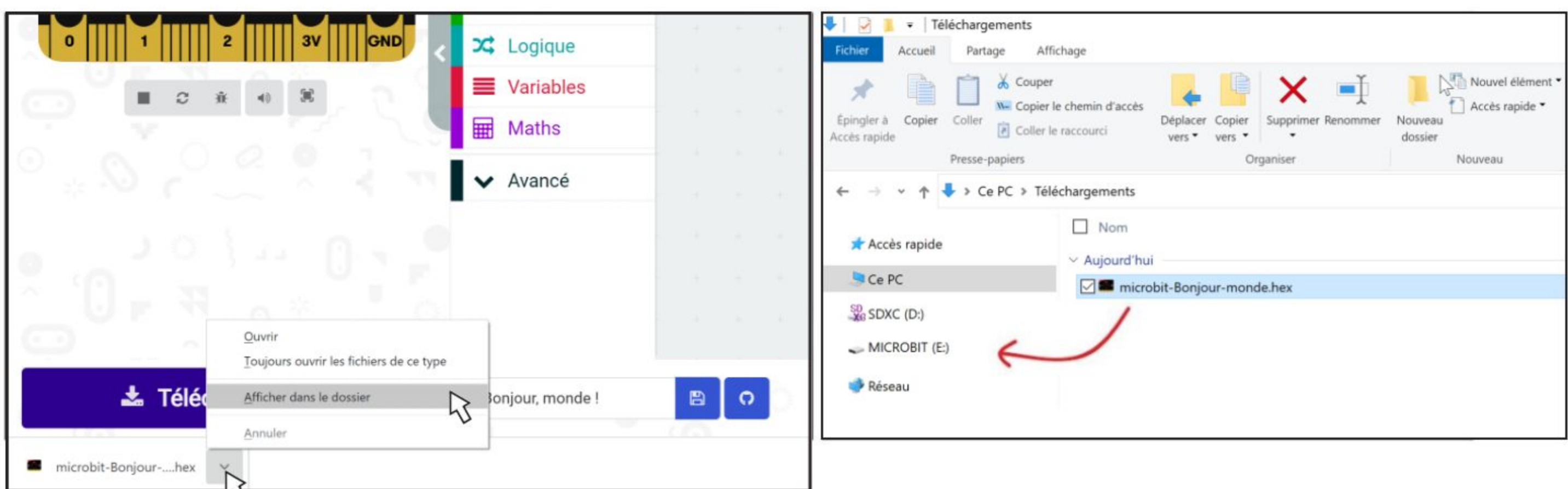


Le processus de téléchargement du code est appelé **Flashage**. La LED orange à l'arrière du micro:bit clignote pendant le transfert, et une fois celui-ci terminé, le code est automatiquement exécuté.

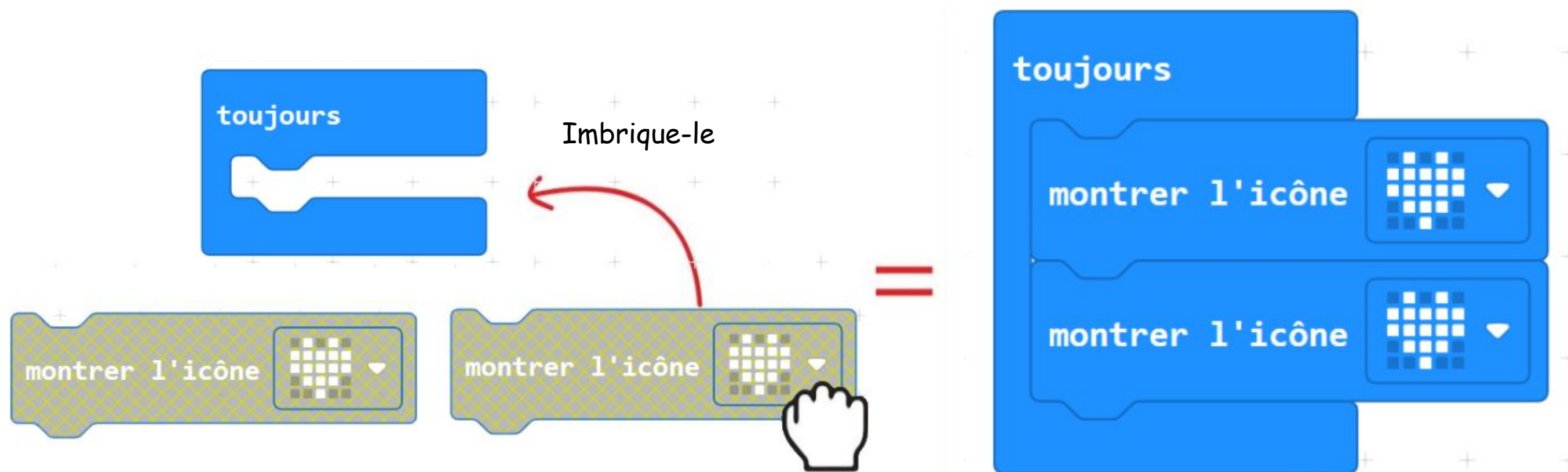


NOTE!

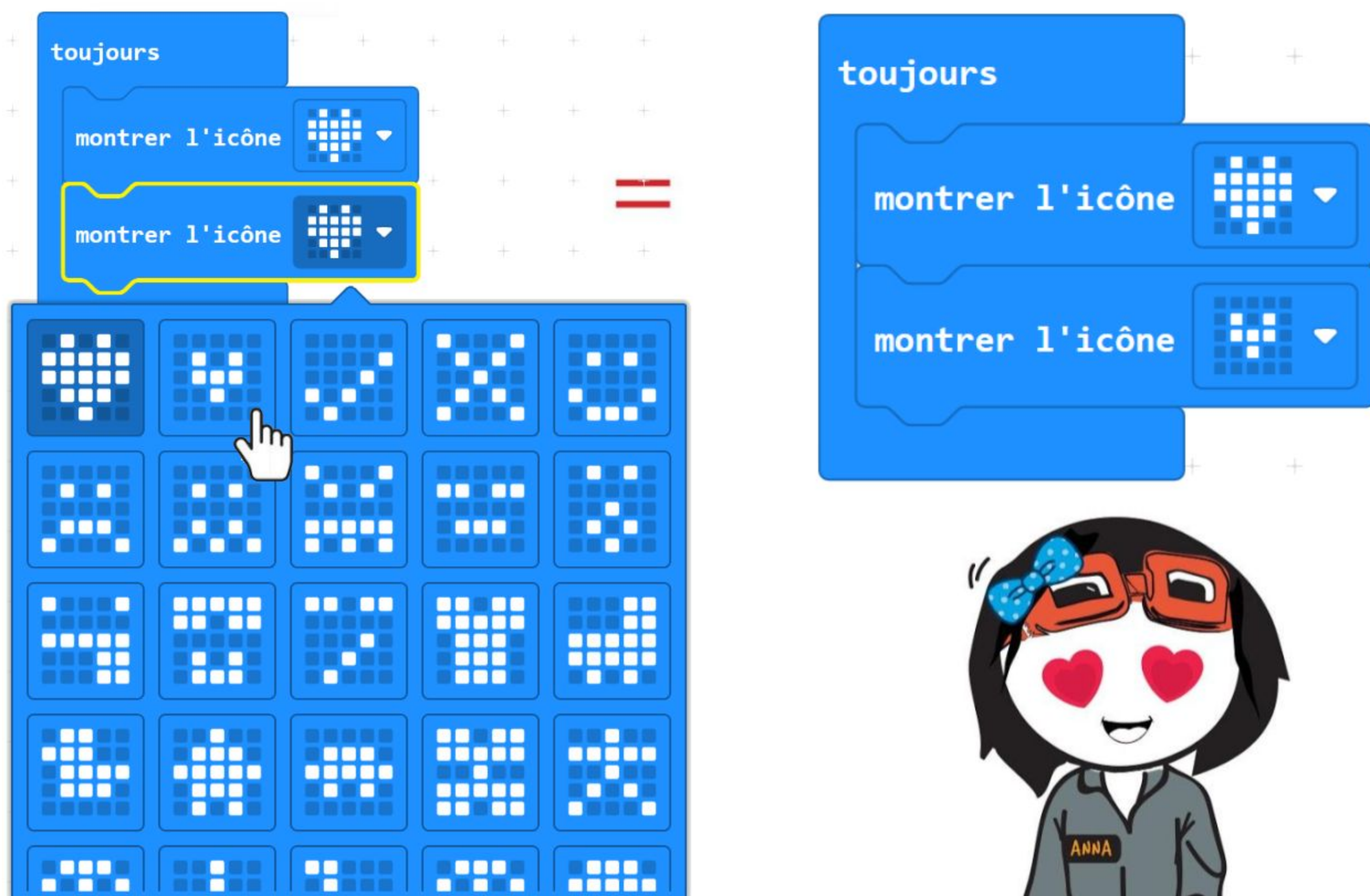
Si aucune fenêtre ne s'affiche, cela signifie que le fichier a été automatiquement téléchargé dans le dossier de téléchargements de ton navigateur. Fais un clic droit sur le fichier .hex téléchargé qui apparaît en bas de la fenêtre, et sélectionne “Afficher dans le dossier”. Clique et maintiens le clic sur le fichier “microbit-xxxx.hex” et dépose-le sur le lecteur MICROBIT, comme si tu déposais un fichier sur une clé USB.



Étape 6 Clique sur [Base], puis clique sur le bloc [montrer l'icône]. Répète l'opération pour obtenir un second bloc [montrer l'icône]. Fais glisser les deux blocs [montrer l'icône] pour les imbriquer dans le bloc [toujours].



Étape 7 Clique sur l'icône du deuxième bloc [montrer l'icône] et sélectionne l'image représentant un petit cœur (le nom "petit cœur" apparaît en infobulle). Flashe le code sur EDU:BIT.



C'est un coup de foudre ! Vois-tu l'animation d'un cœur qui bat ?

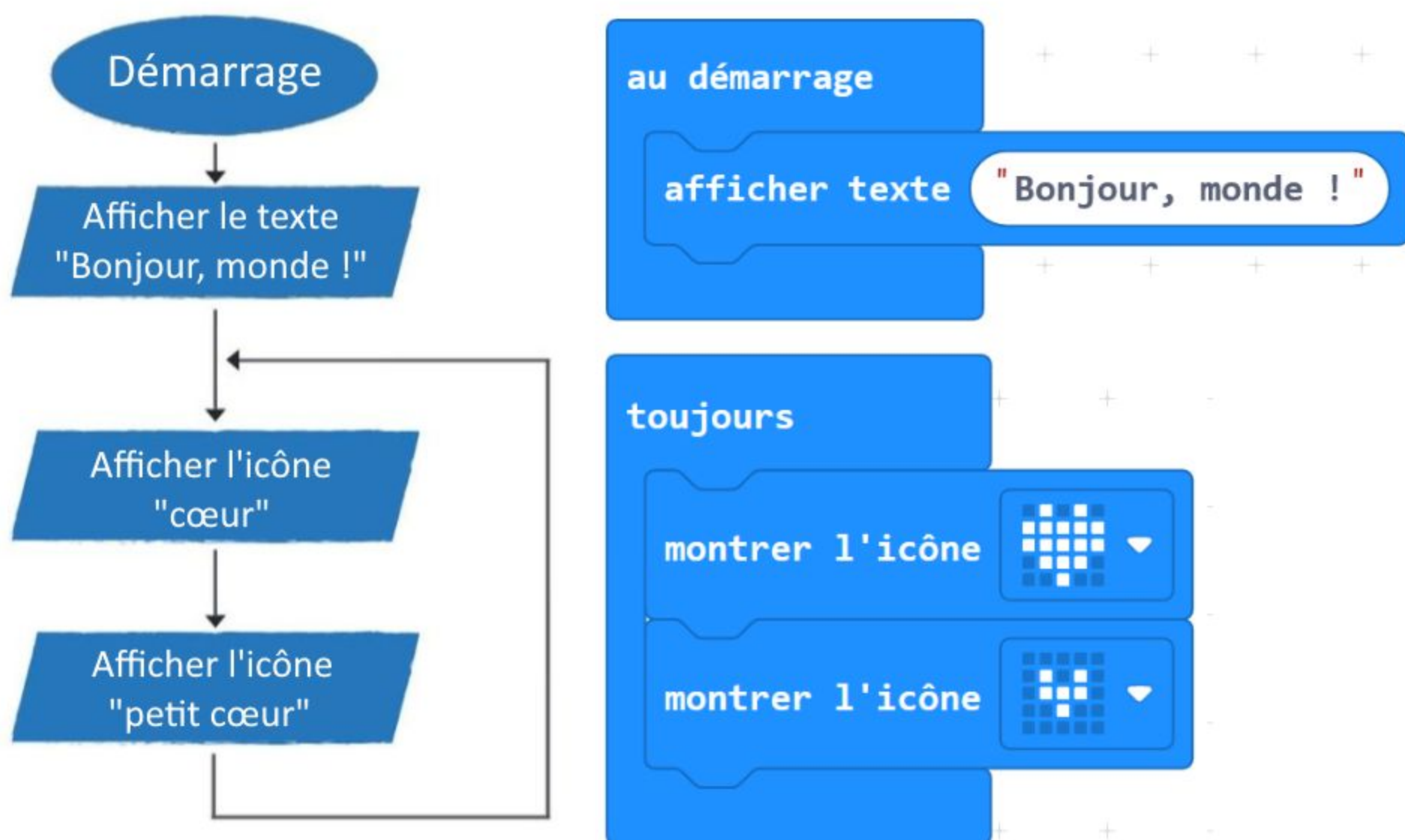


DÉCHIFFRE

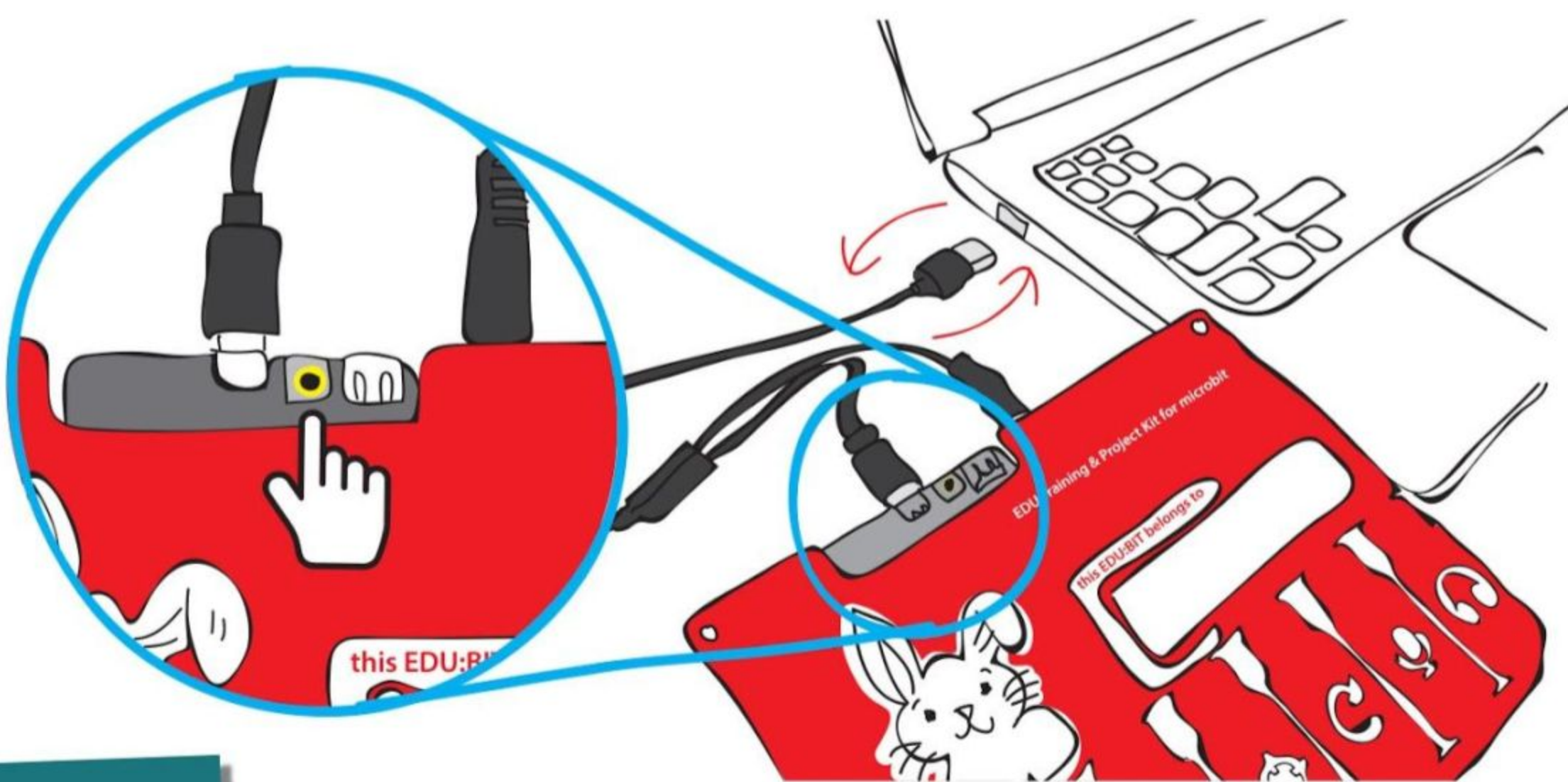
LE CODE



As-tu remarqué que le texte "Bonjour, monde !" ne défile qu'une seule fois, alors que l'animation du cœur tourne en boucle ? Pourquoi ?



Le bloc [**au démarrage**] est exécuté une seule et unique fois au démarrage, tandis que le bloc [**toujours**] est constamment répété.

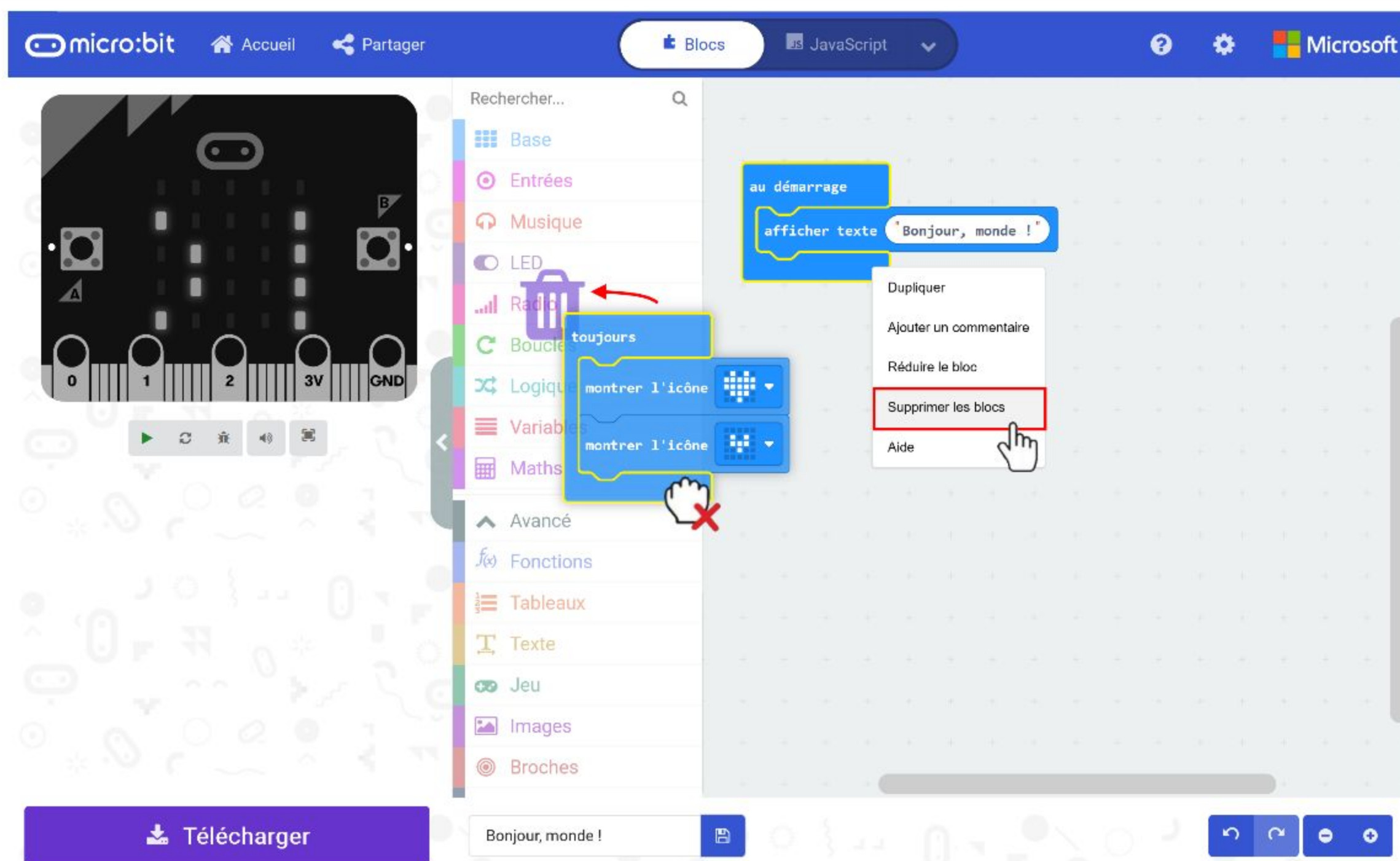


NOTE !

Si tu souhaites relancer le programme au début, appuie simplement sur le bouton RESET à l'arrière de la carte, ou débranche puis rebranche le câble USB.

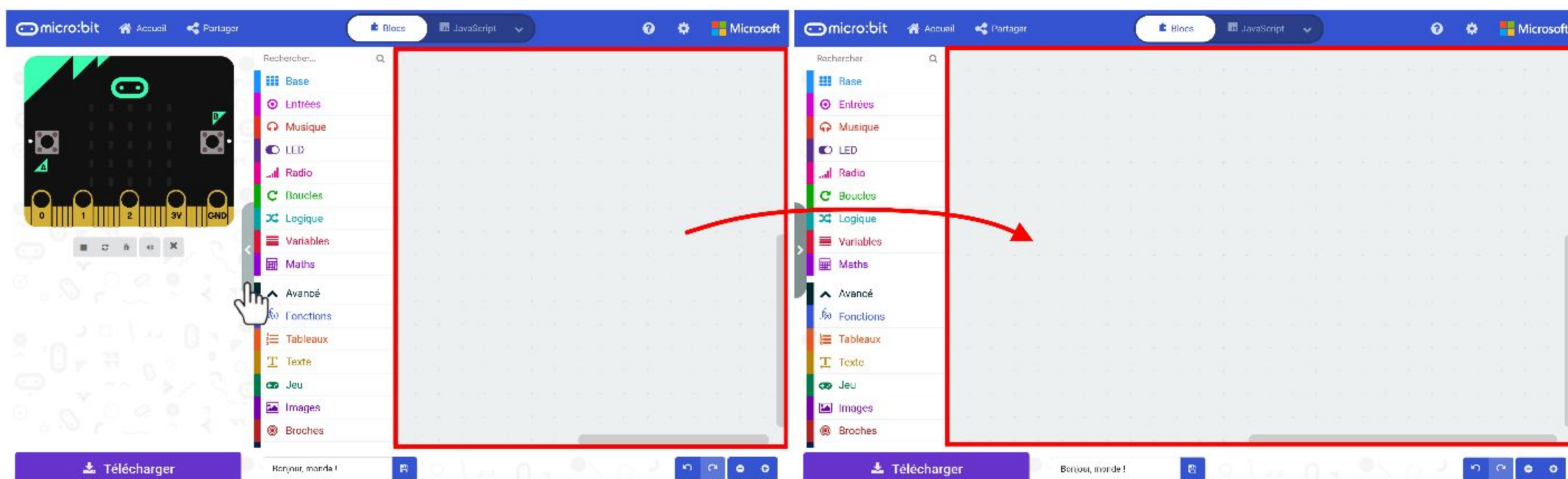
Trucs & Astuces #1 !

Tu peux effacer un ou des blocs en le(s) faisant glisser vers la boîte à outils. Une icône représentant une poubelle apparaît alors. Tu peux également faire un clic droit sur le(s) bloc(s) à supprimer et sélectionner "Supprimer le(s) bloc(s)".



Trucs & Astuces #2 !

Si tu n'utilises pas le simulateur, tu peux le cacher en cliquant sur l'onglet comme montré ci-dessous. Ainsi, tu auras plus d'espace pour tes blocs de codage.





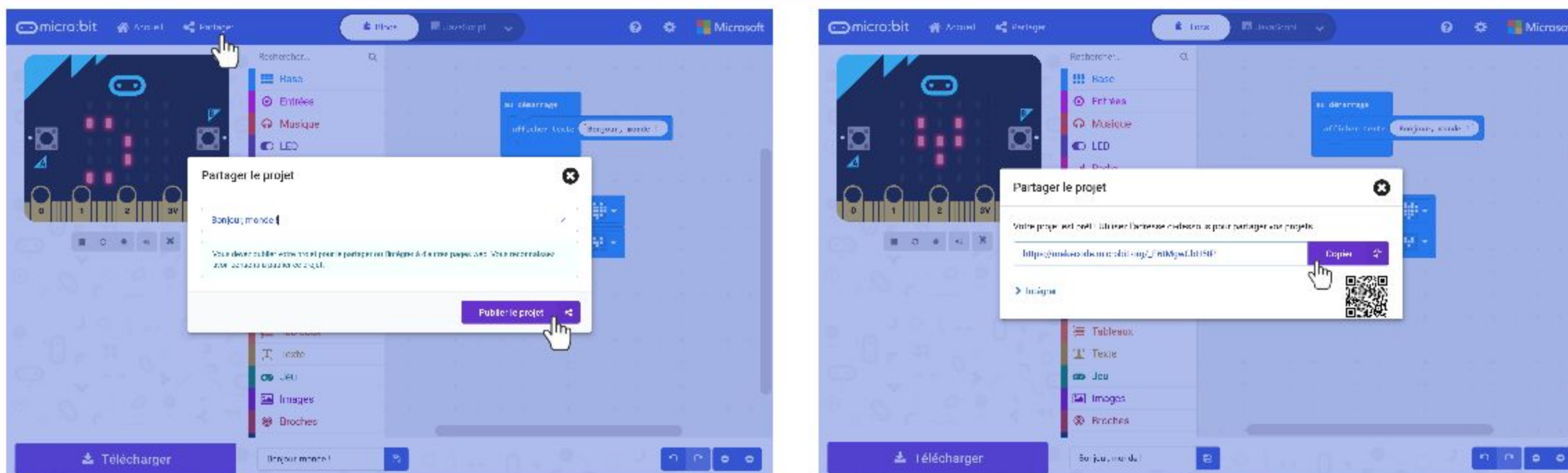
Trucs & Astuces #3 !

Tu peux réduire un bloc qui en contient d'autres en faisant un clic droit dessus et en sélectionnant "Réduire le bloc". Pour le redévelopper, il te suffit de cliquer sur l'icône .



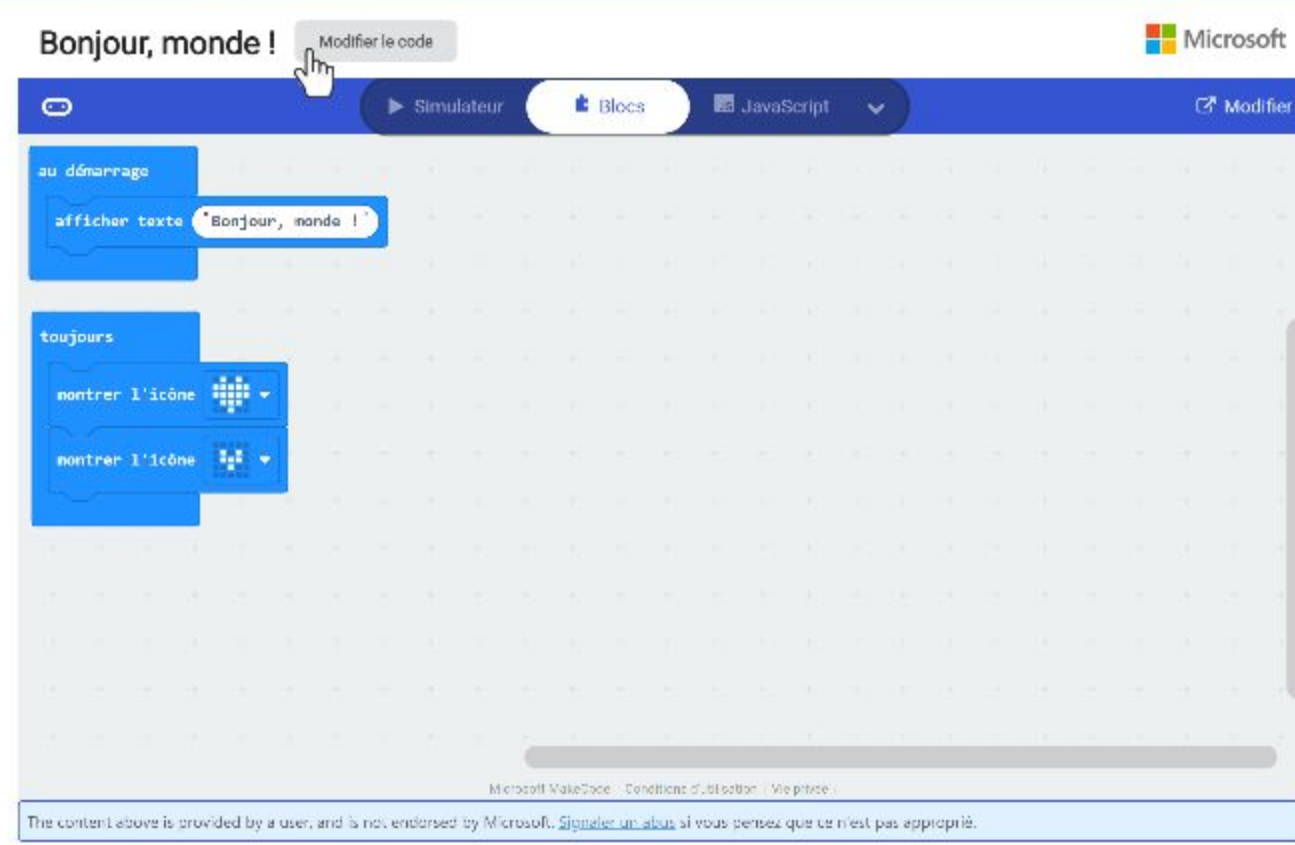
Trucs & Astuces #4 !

Tu peux partager ton code avec ton professeur ou tes amis en publiant ton projet et en leur envoyant l'URL du projet. Pour cela, clique sur [**Partager**], puis sur [**Publier le projet**] dans la fenêtre pop-up. Une autre fenêtre pop-up s'ouvre alors ; elle contient l'URL de ton projet. Clique sur le bouton [**Copier**] pour copier l'URL du projet.



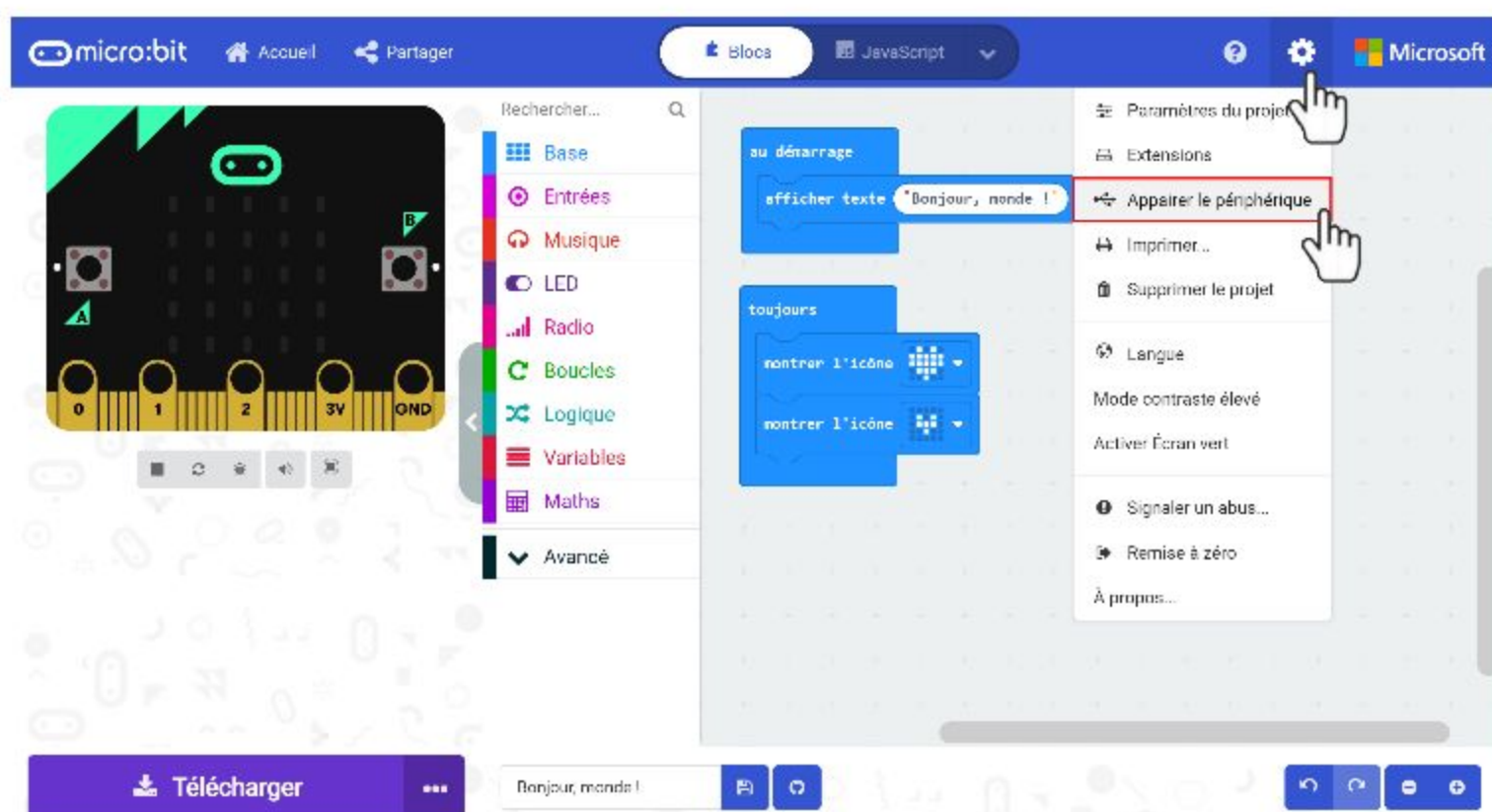
Trucs & Astuces #5 !

Ton professeur ou tes amis verront la page ci-dessous lorsqu'ils ouvriront l'URL de ton projet dans un navigateur internet. Ils peuvent voir ton code, mais aussi le modifier en cliquant sur le bouton [**Modifier le code**].



Trucs & Astuces #6 !

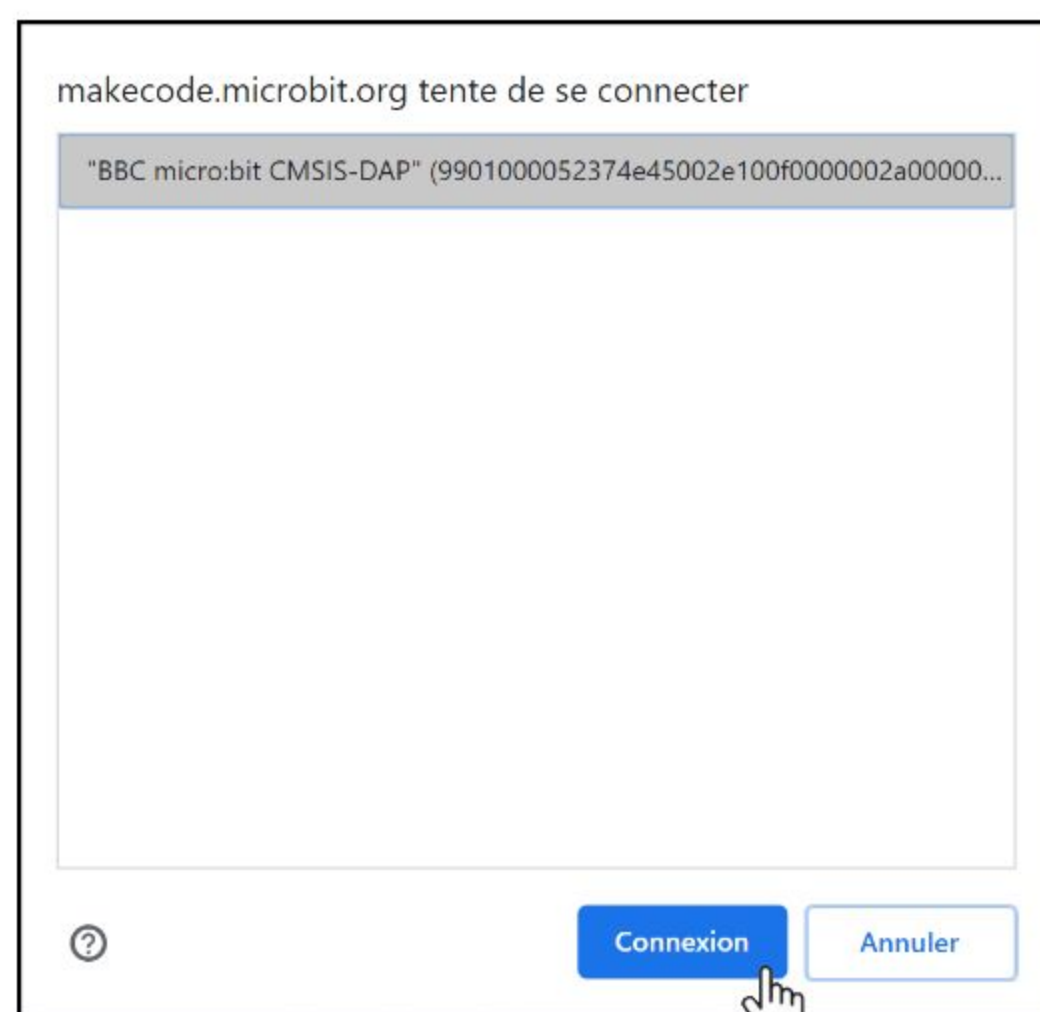
Sais-tu que tu peux appairer un périphérique pour télécharger tes programmes en un clic ? Pour cela, clique sur la roue dentée, puis sur "Appairer le périphérique".



NOTE !

Tu dois utiliser la dernière version de Google Chrome ou de Microsoft Edge. Par ailleurs, le firmware de ta carte micro:bit doit être à jour. Pour mettre ton micro:bit à jour, suis les instructions du lien suivant : <https://microbit.org/get-started/user-guide/firmware/>

Assure-toi qu'EDU:BIT est connecté à ton PC et clique sur le bouton [**Appairer le périphérique**] dans la fenêtre pop-up. Ensuite, sélectionne BBC micro:bit CMSIS-DAP ou DAPLink CMSIS-DAP dans la liste puis clique sur [**Connecter**].



Une fois l'appairage effectué, tu peux directement flasher ton code sur EDU:BIT en cliquant sur le bouton [**Télécharger**]. Essaie !

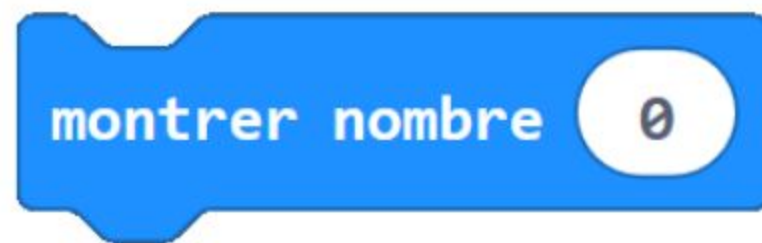
Si tu rencontres un problème lors de l'appairage, tu peux consulter le lien ci-dessous pour plus d'informations :
<https://makecode.microbit.org/device/usb/webusb/troubleshoot>





EXPLORE D'AUTRES BLOCS

#1 Utilise le bloc [**montrer LEDs**] pour créer tes propres icônes et le bloc [**montrer nombre**] pour afficher un nombre.



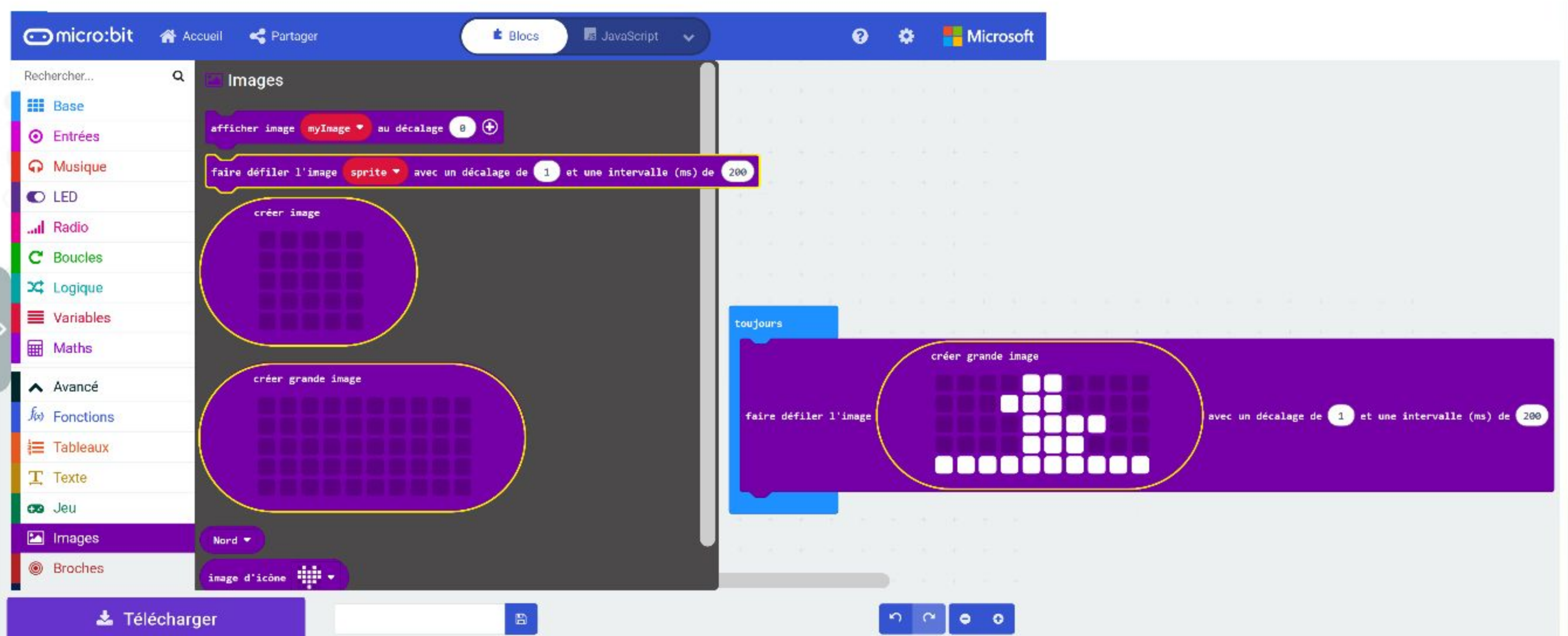
Le savais-tu ?

1 000 millisecondes (ms) = 1 seconde

#2 Ajoute un bloc [**pause _ (ms)**] pour ralentir ton programme. Cette fonction met le programme en pause pendant la durée indiquée en millisecondes (ms).



#3 Pour faire défiler une image sur la matrice de LEDs, tu peux utiliser le bloc [**faire défiler l'image _ avec un décalage de _ et un intervalle (ms)**], combiné avec le bloc [**créer image**] ou le bloc [**créer grande image**], de la catégorie [**Images**] (sous la catégorie **Avancé**, qu'il faut développer).



Lorsque tu exécutes ce programme, tu peux voir des petits canetons défiler l'un après l'autre, sur la matrice de LEDs.

APPLICATION DÉFI

Programme EDU:BIT pour qu'elle serve de tableau d'affichage.

au démarrage

Afficher une animation simple afin d'attirer l'attention, puis afficher le nom de ta classe.

toujours

Afficher la date et les informations importantes pour la classe.



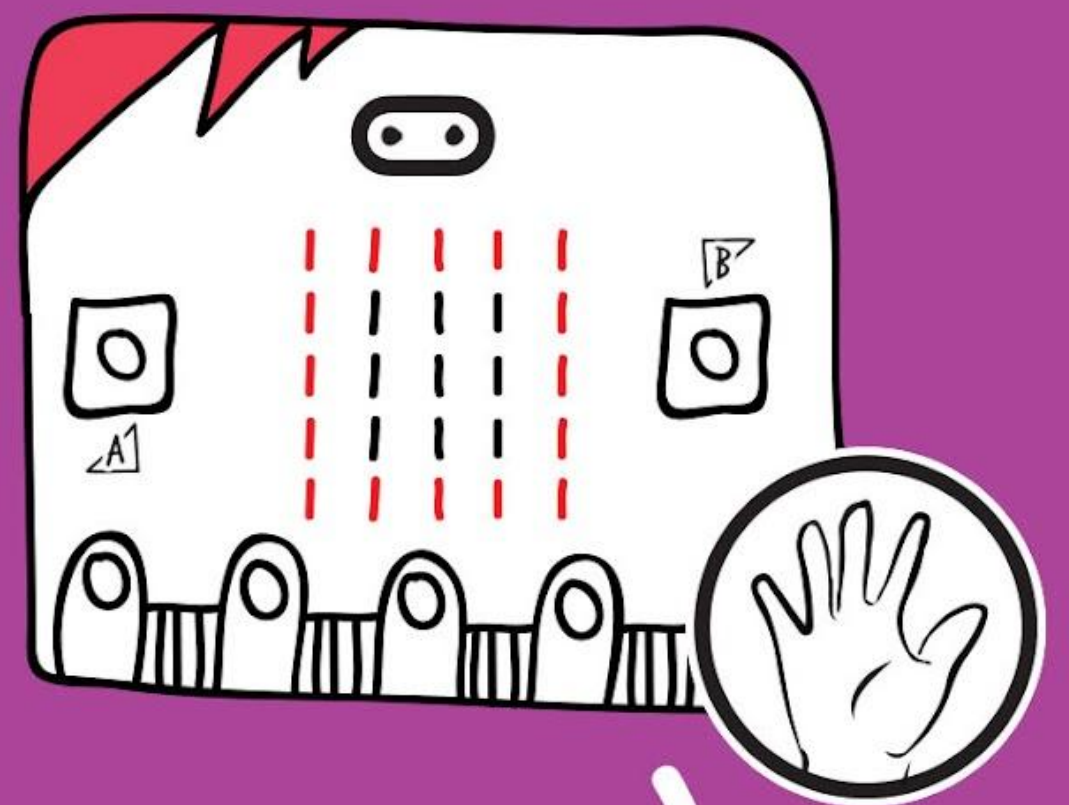
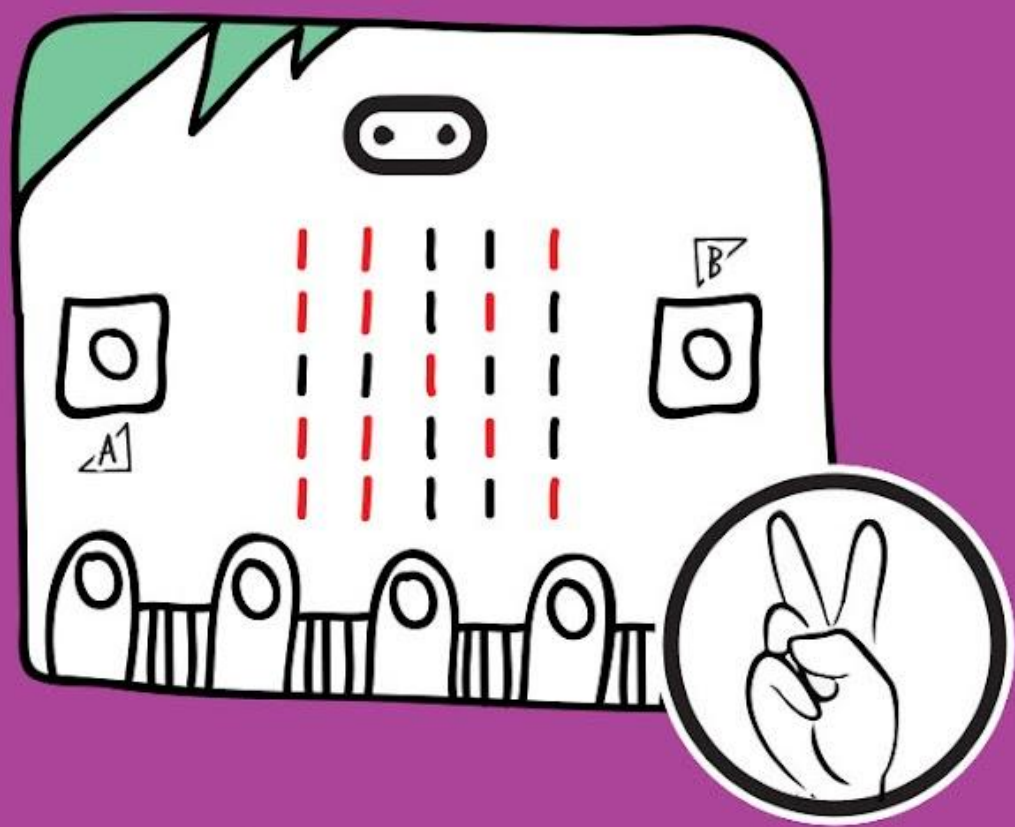
CHAPITRE 2

Jouons à Pierre, Papier, Ciseaux !

Appuie sur les boutons du micro:bit et du Button Bit

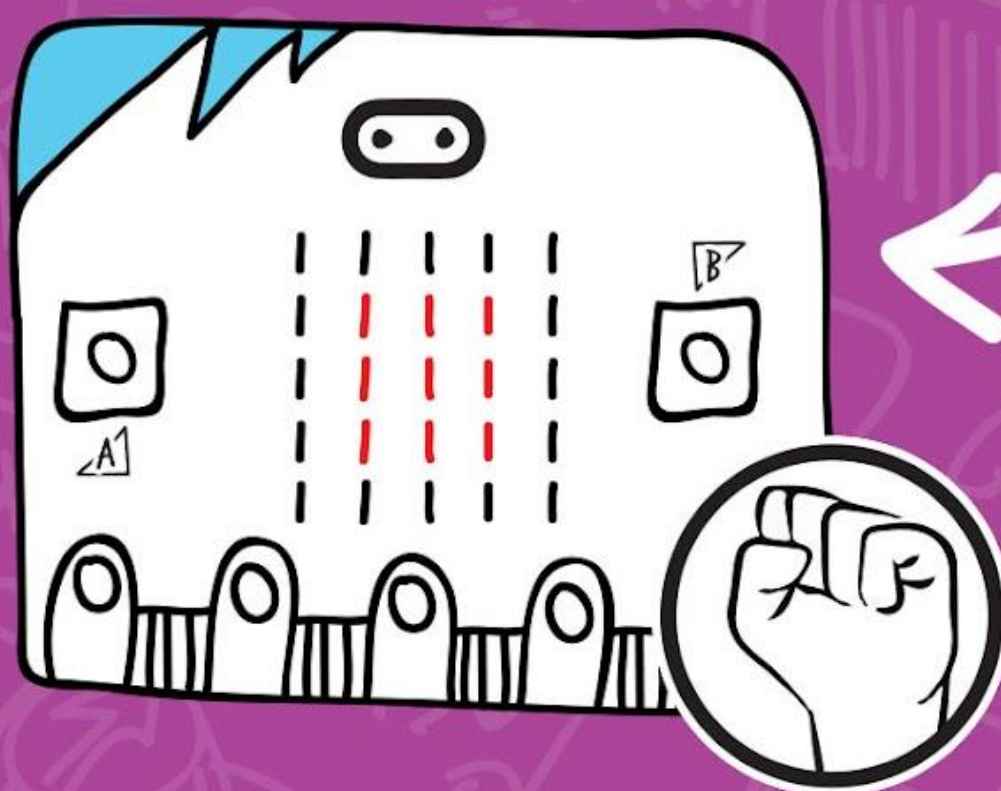
Ciseaux

Battent Papier



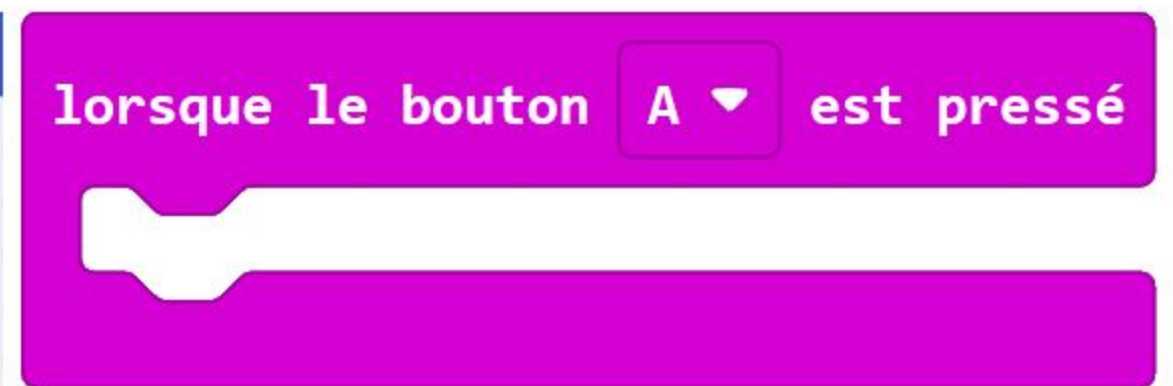
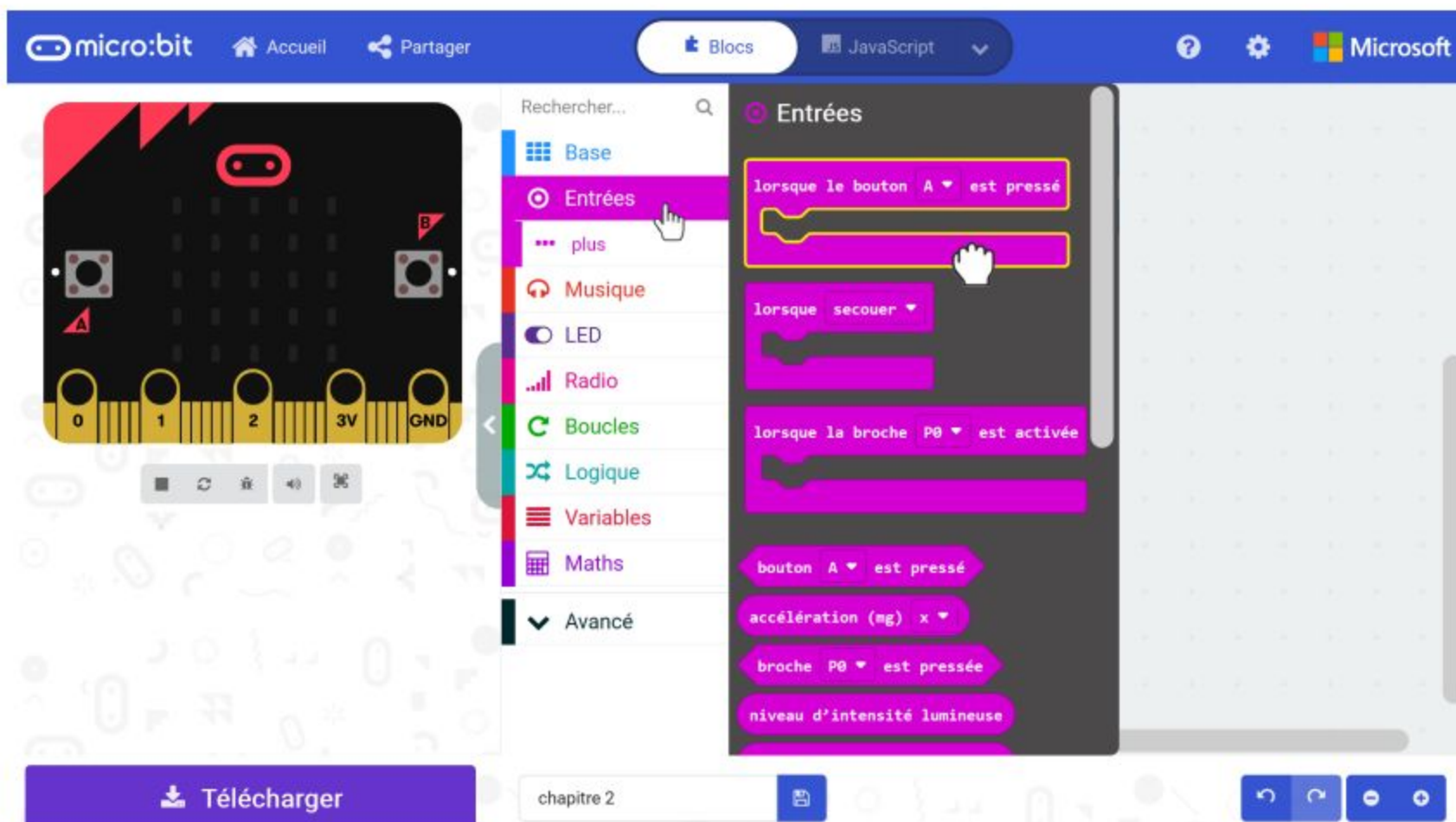
Pierre
Bat Ciseaux

Papier
Bat Pierre

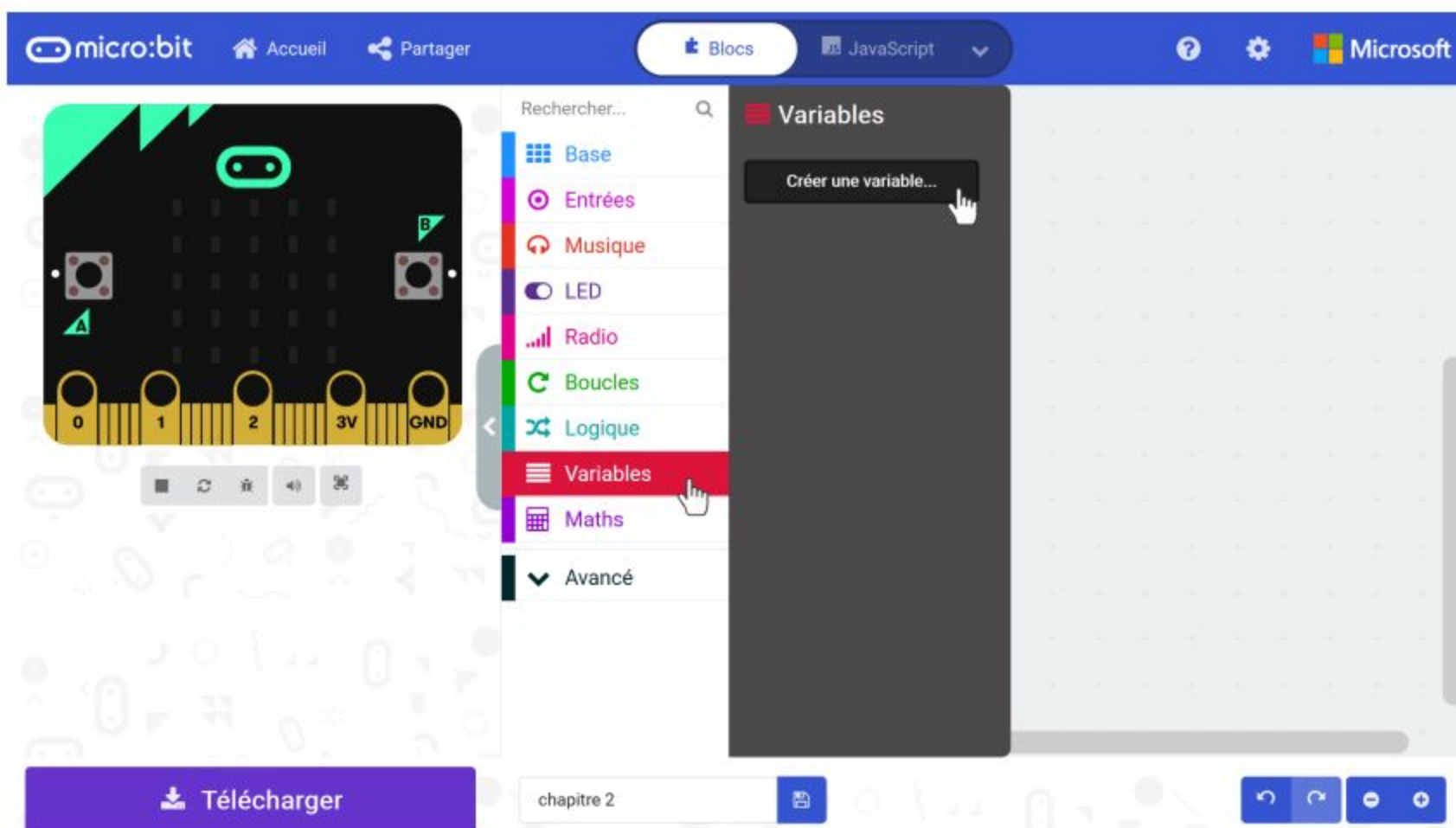


CODONS !

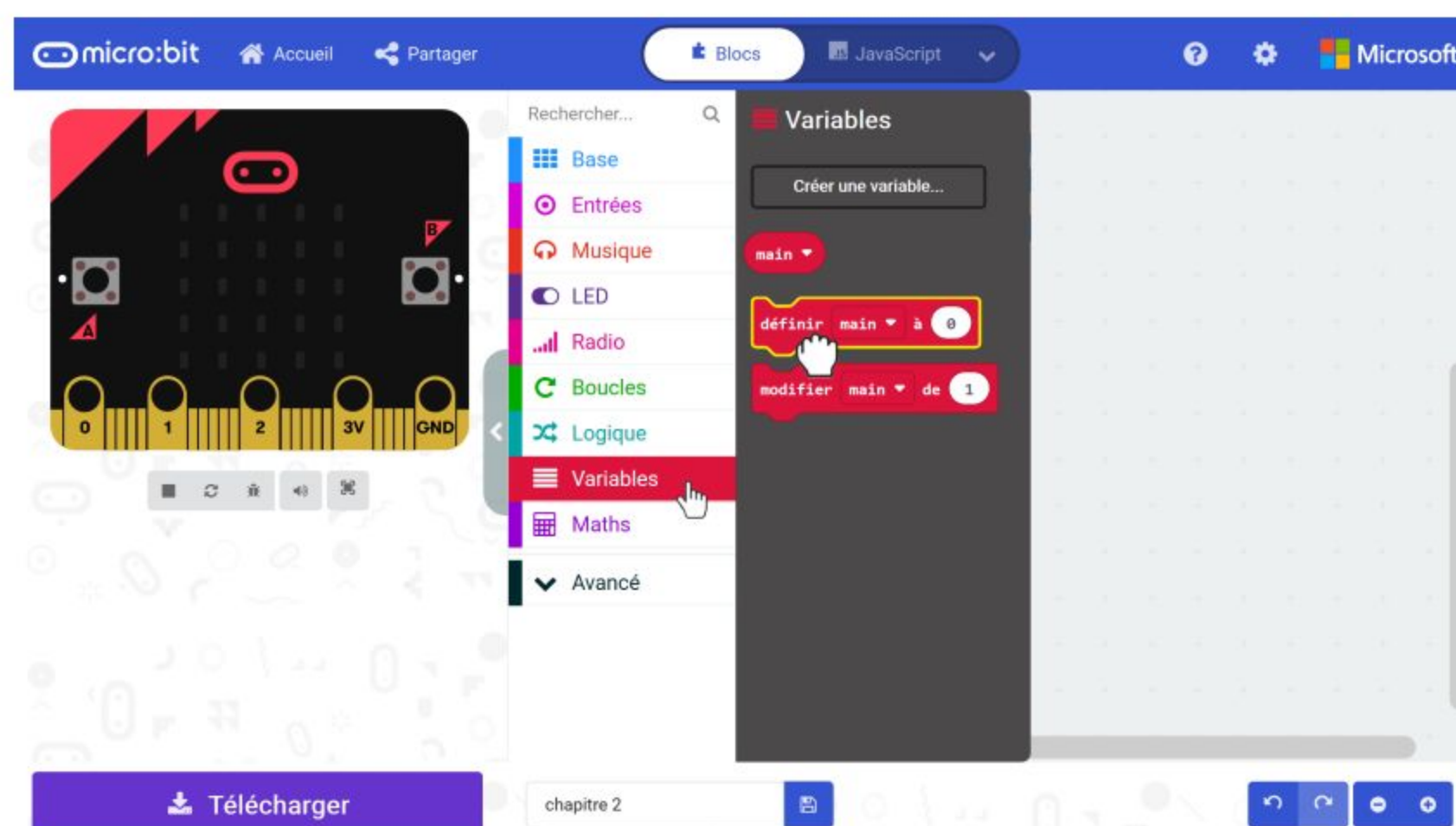
Étape 1 Vas sur <https://makecode.microbit.org/> (ou clique simplement sur l'icône **Accueil** si tu es déjà dans l'éditeur MakeCode) et crée un nouveau projet. Clique sur la catégorie **[Entrées]** et sélectionne le bloc **[lorsque le bouton _ est pressé]**.



Étape 2 Clique sur la catégorie **[Variables]** puis clique sur **[Créer une variable...]**. Tape "main" dans le champ de saisi de la fenêtre pop-up. Valide en cliquant sur OK.

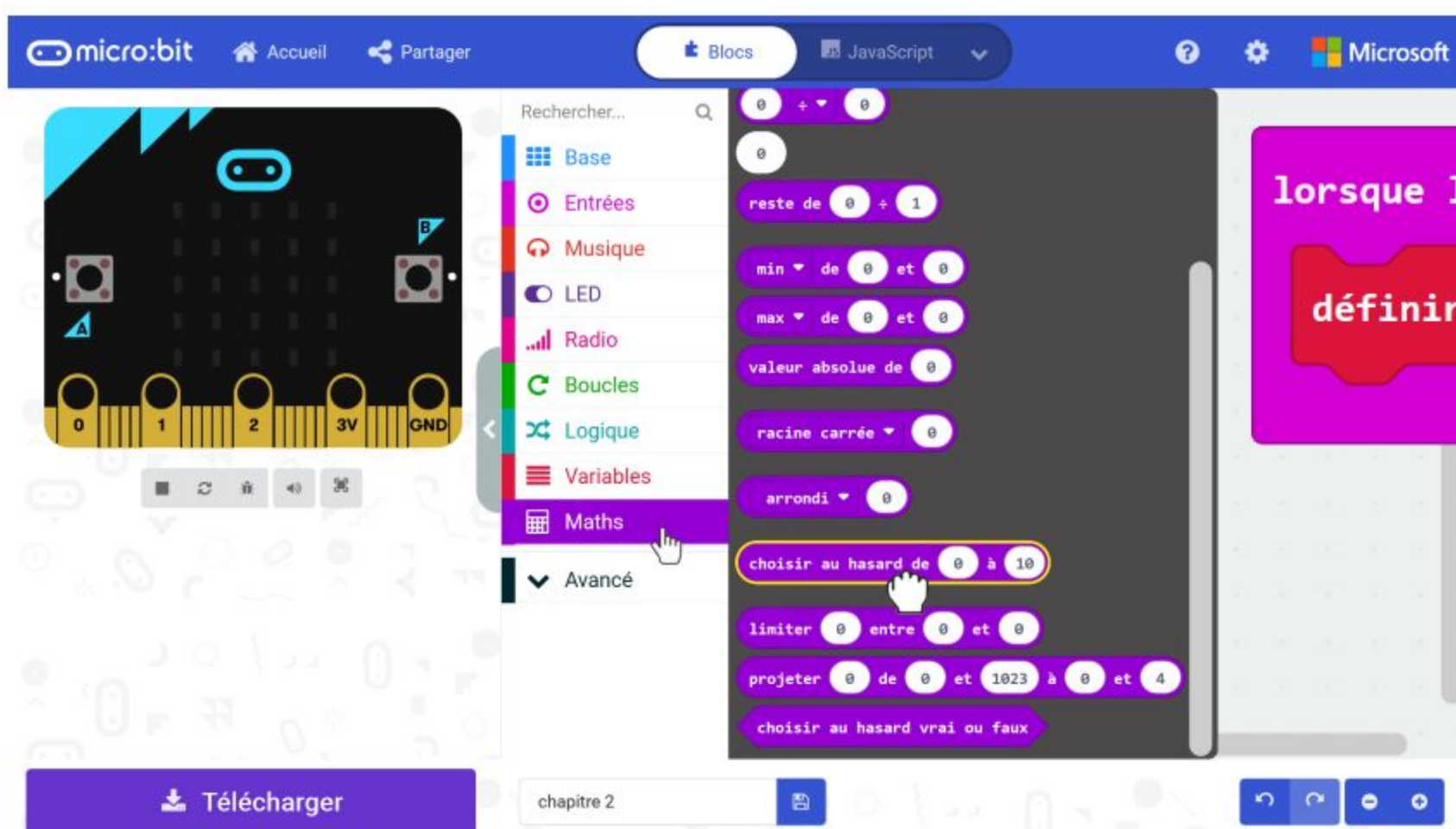


Étape 3 Clique sur la catégorie **[Variables]** et sélectionne le bloc **[définir _ à _]**. Imbrique-le dans le bloc **[lorsque le bouton _ est pressé]**.

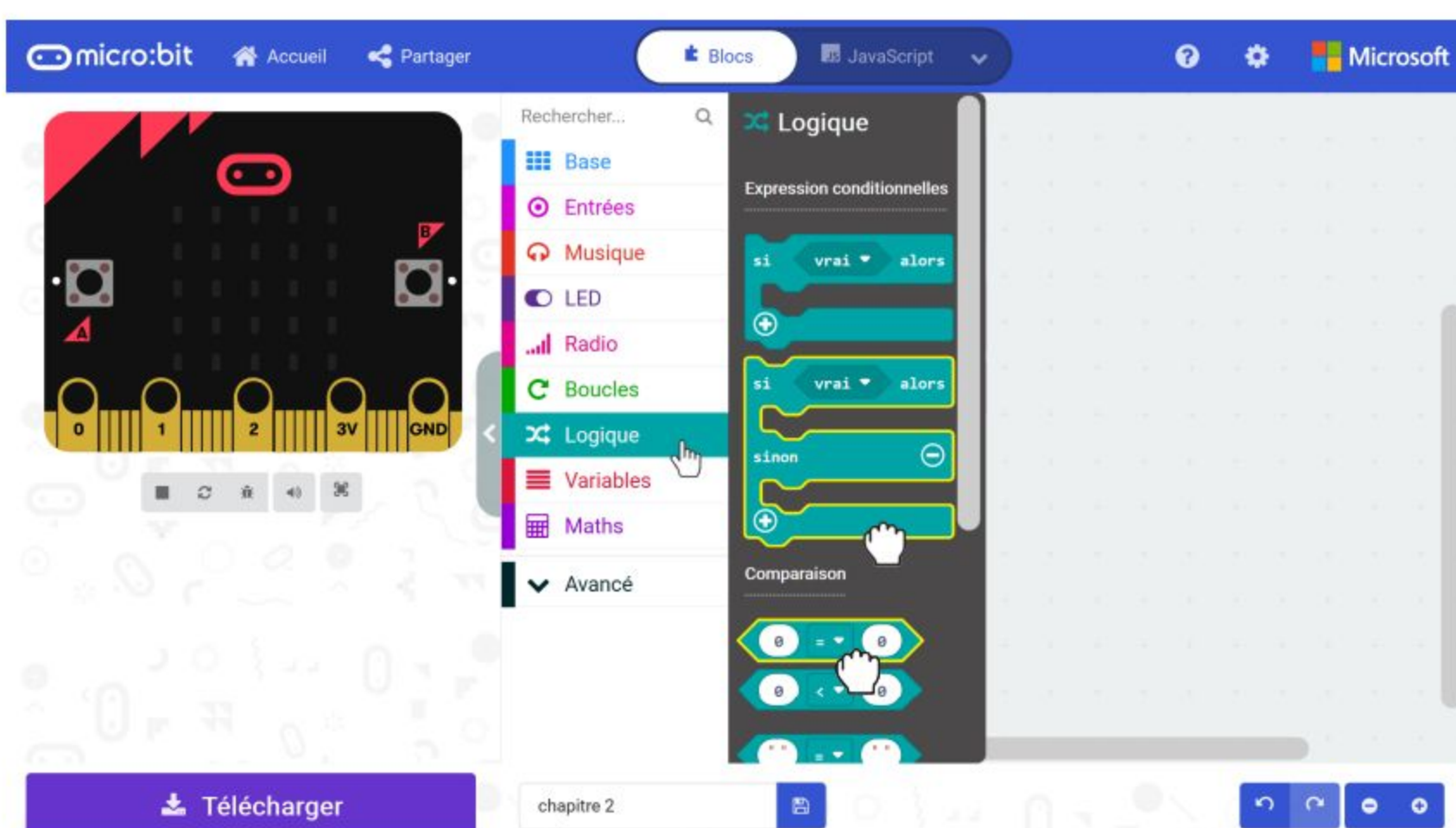




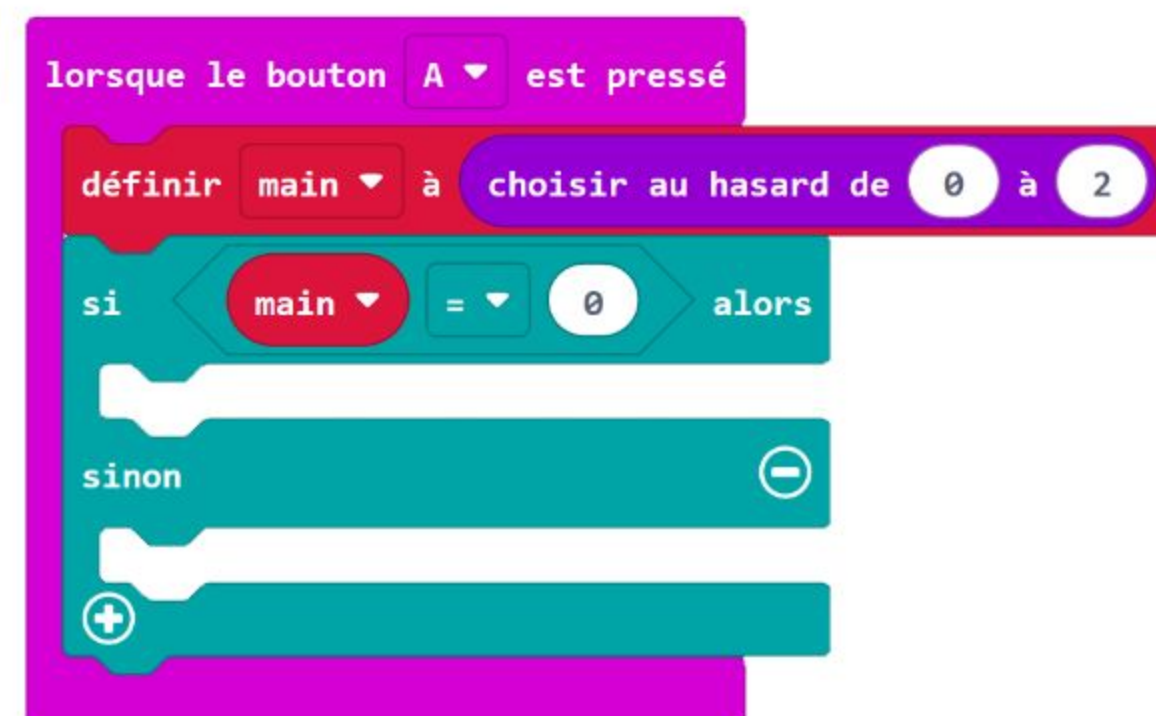
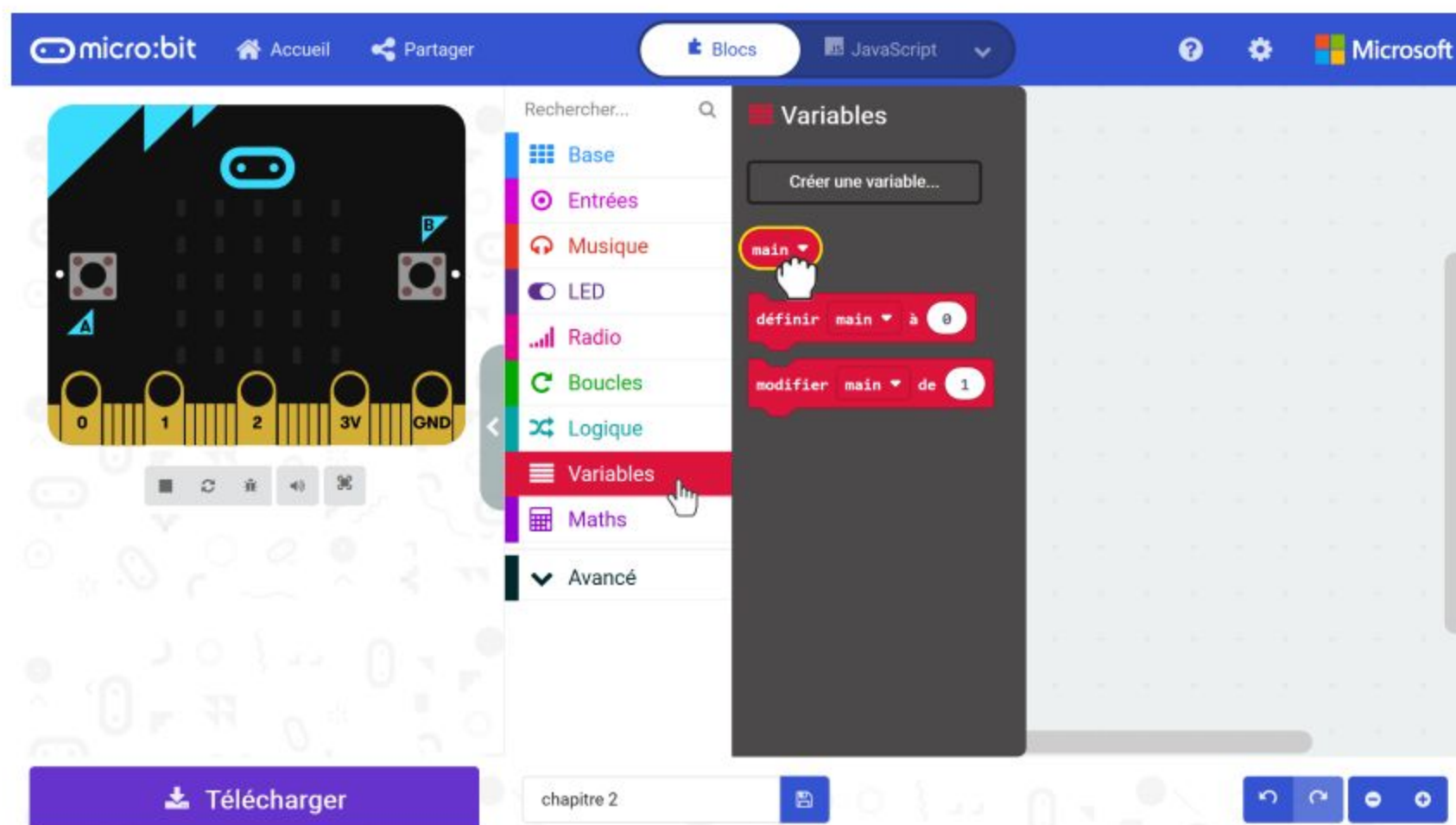
Étape 4 Clique sur la catégorie **[Maths]** et sélectionne le bloc **[choisir au hasard de _ à _]**. Remplace le nombre 10 par 2.



Étape 5 Clique sur la catégorie **[Logique]**, sélectionne le bloc **[si-alors-sinon]** et le bloc de comparaison **[_ = _]**. Insère le bloc **[_ = _]** dans la condition du "si".

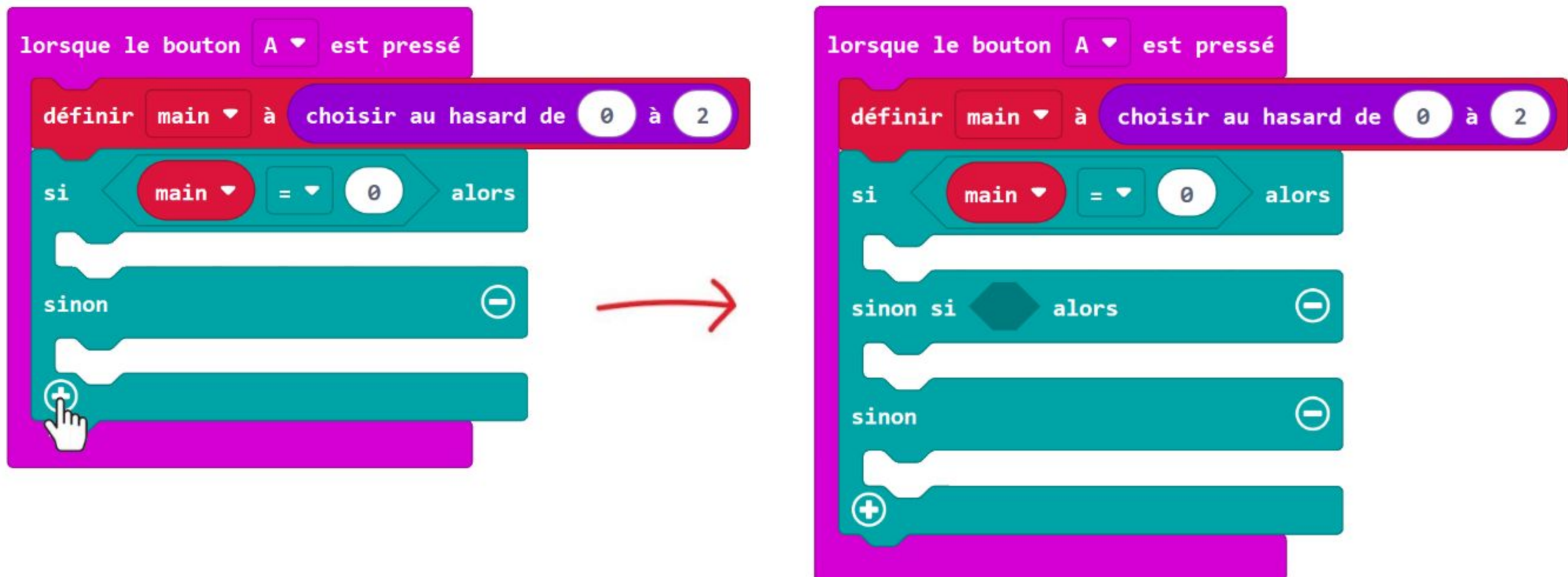


Étape 6 Clique sur la catégorie **[Variables]** et sélectionne le bloc **[main]**. Insère-le dans le premier membre du bloc de comparaison.

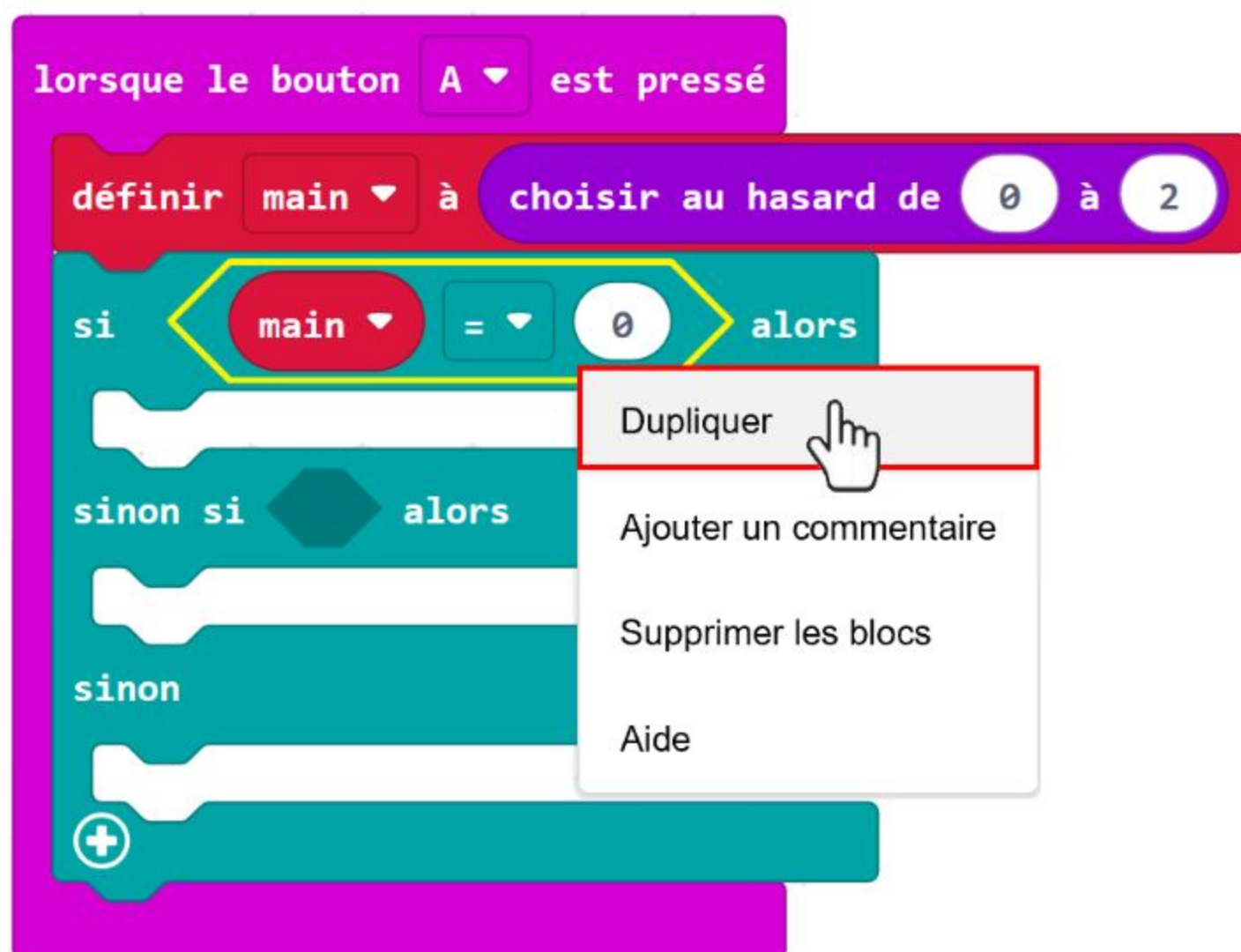


CHAPITRE 2 : Jouons à Pierre, Papier, Ciseaux !

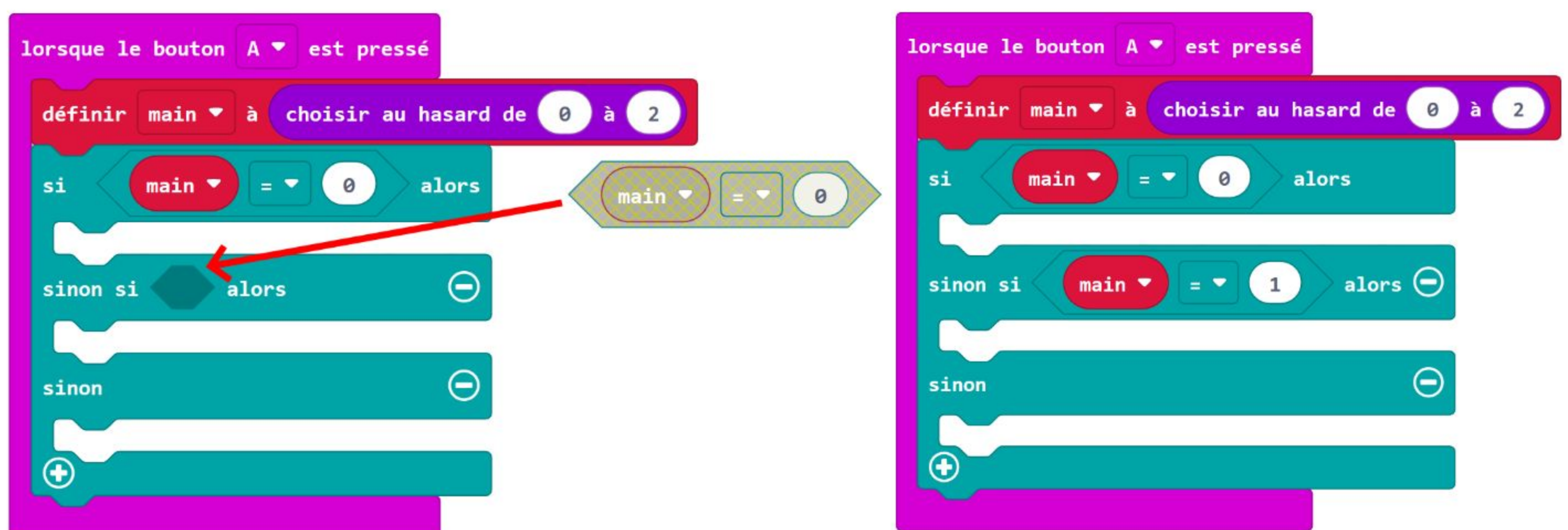
Étape 7 Clique sur l'icône **+** pour ajouter un test **[sinon si]** dans le bloc "si".



Étape 8 Fais un clic droit sur le bloc de comparaison et sélectionne "Dupliquer".



Étape 9 Insère le bloc dupliqué dans la condition du test "sinon si" et remplace le nombre 0 par 1.





Étape 10 Imbrique des blocs [Base] : [montrer LEDs] dans les emplacements “si”, “sinon si” et “else”. Clique sur les pixels des blocs [montrer LEDs] pour créer les images comme ci-dessous.

Essayons !



lorsque le bouton A est pressé

définir main à choisir au hasard de 0 à 2

si main = 0 alors

montrer LEDs

PIERRE

sinon si main = 1 alors

montrer LEDs

PAPIER

sinon

montrer LEDs

CISEAUX

Flashe le code sur EDU:BIT. Maintenant, tu peux jouer à Pierre, Papier, Ciseaux avec tes amis. Peu importe que tu appuies sur le bouton A du micro:bit ou sur le bouton jaune du Button Bit, la matrice de LEDs affichera au hasard “Pierre”, “Papier” ou “Ciseaux”.

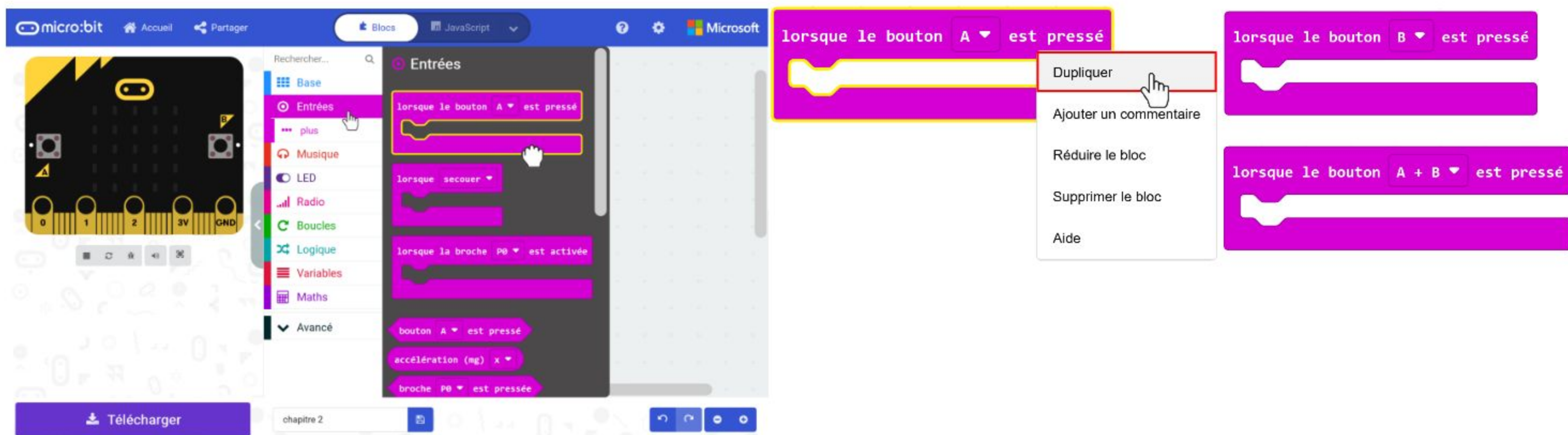
Rappel : Si tu souhaites conserver ton projet, pense à le sauvegarder dans un dossier de ton ordinateur en cliquant sur l’icône “Enregistrer”.



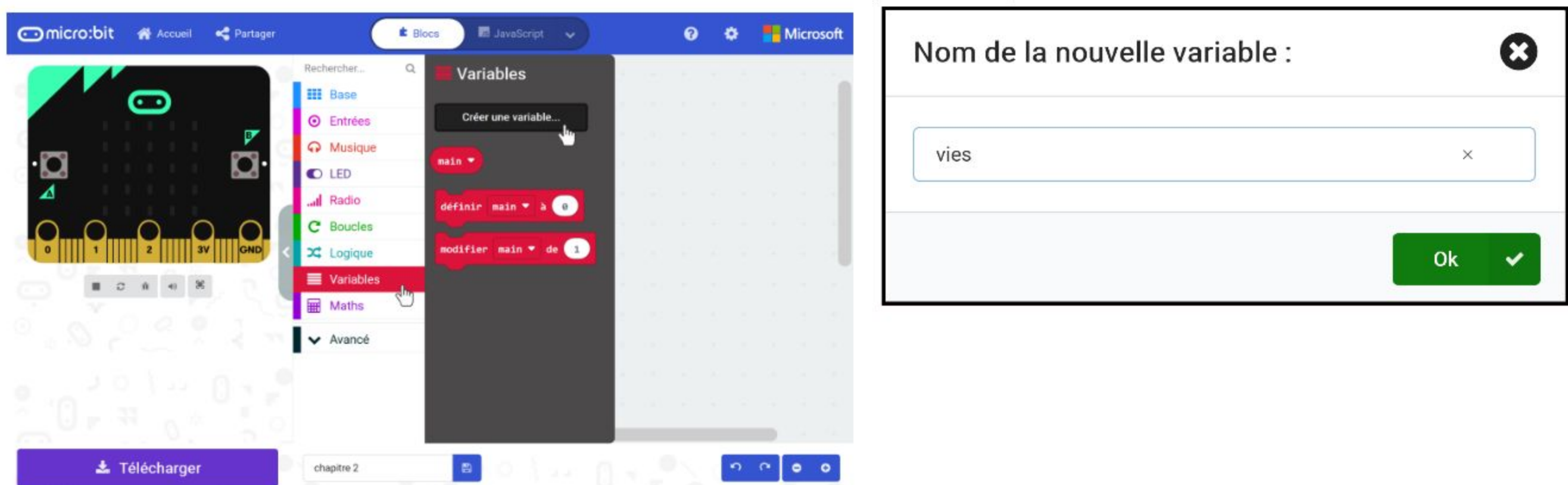


Améliorons le jeu en attribuant 3 vies à chaque joueur. Pour faire ceci, tu dois créer une nouvelle variable nommée "vies" et ajouter les blocs suivants dans ton code.

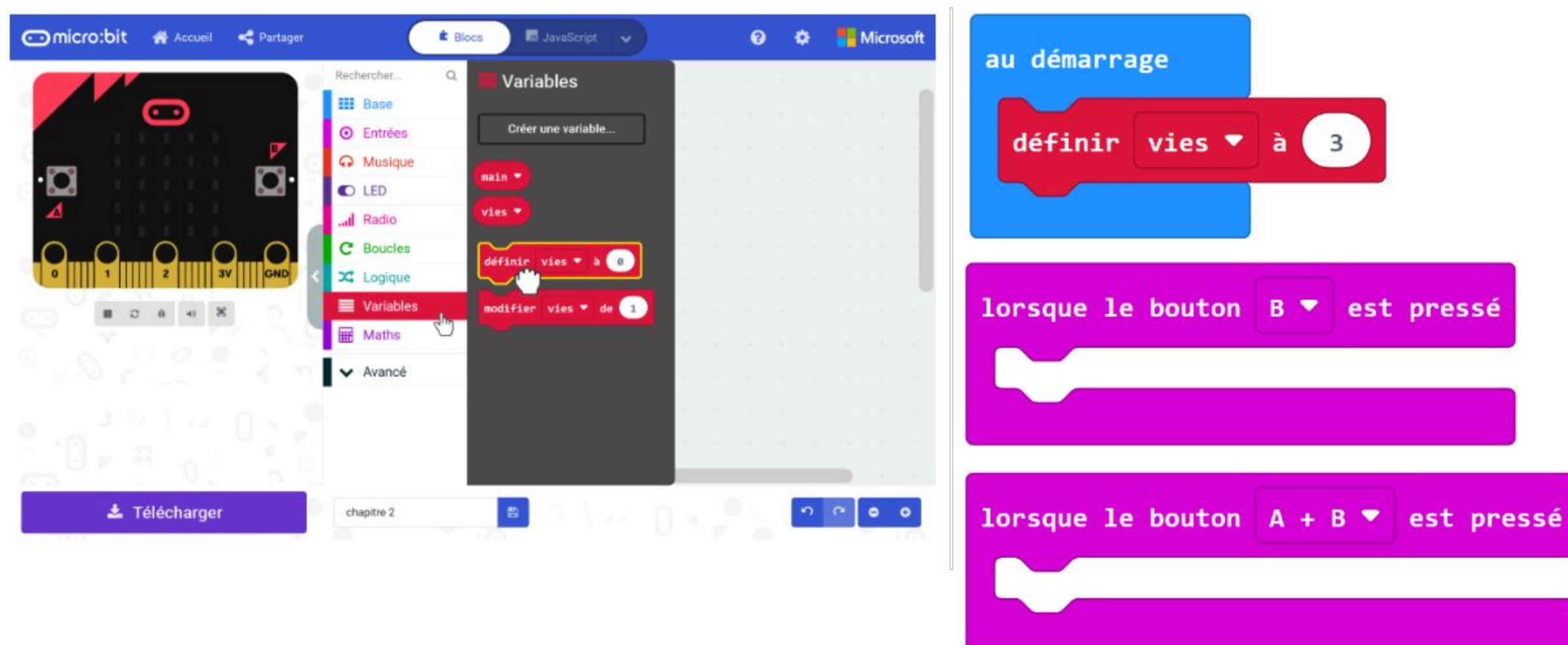
Étape 11 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**]. Duplique-le et change les boutons en "bouton B" et "bouton A+B" respectivement.



Étape 12 Clique sur la catégorie [**Variables**] et sélectionne [**Créer une variable...**]. Tape "vies" dans le champ de saisie de la fenêtre pop-up, puis clique sur OK.

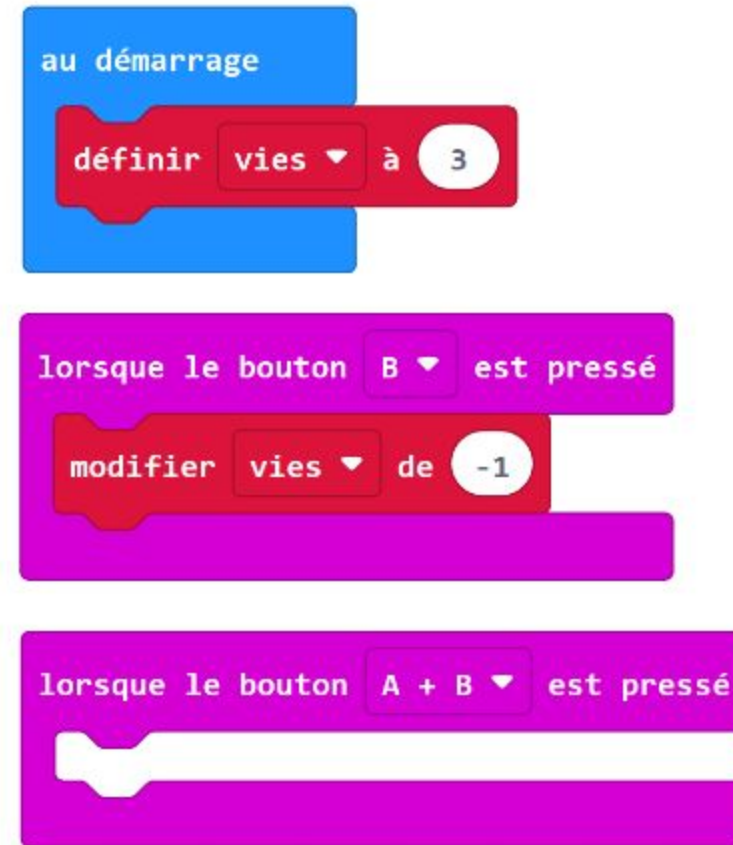
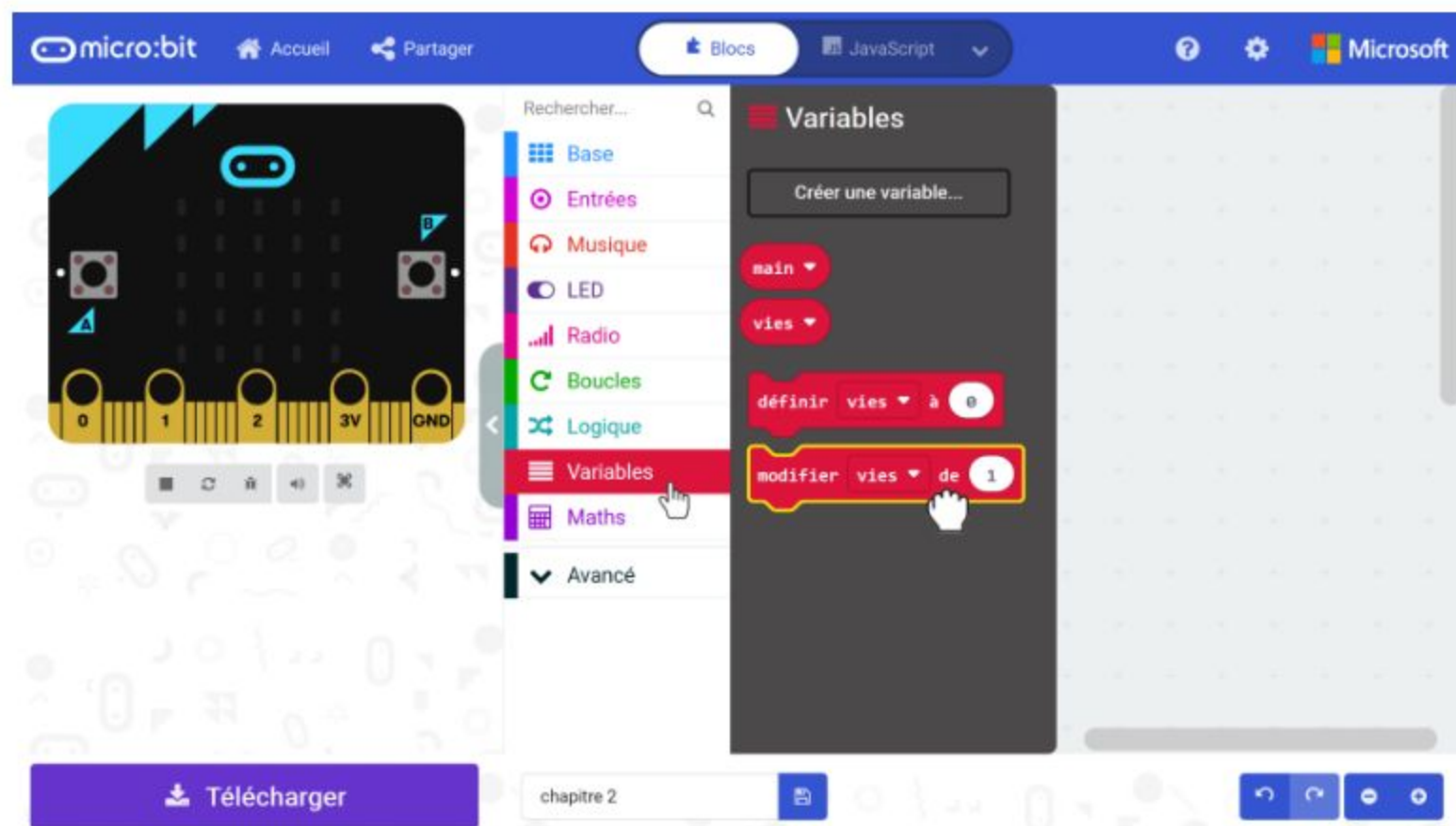


Étape 13 Clique sur la catégorie [**Variables**] et sélectionne le bloc [**définir _ à _**]. Imbrique-le dans le bloc [**Base**] : [**au démarrage**]. Change la variable en "vies" et la valeur en 3.

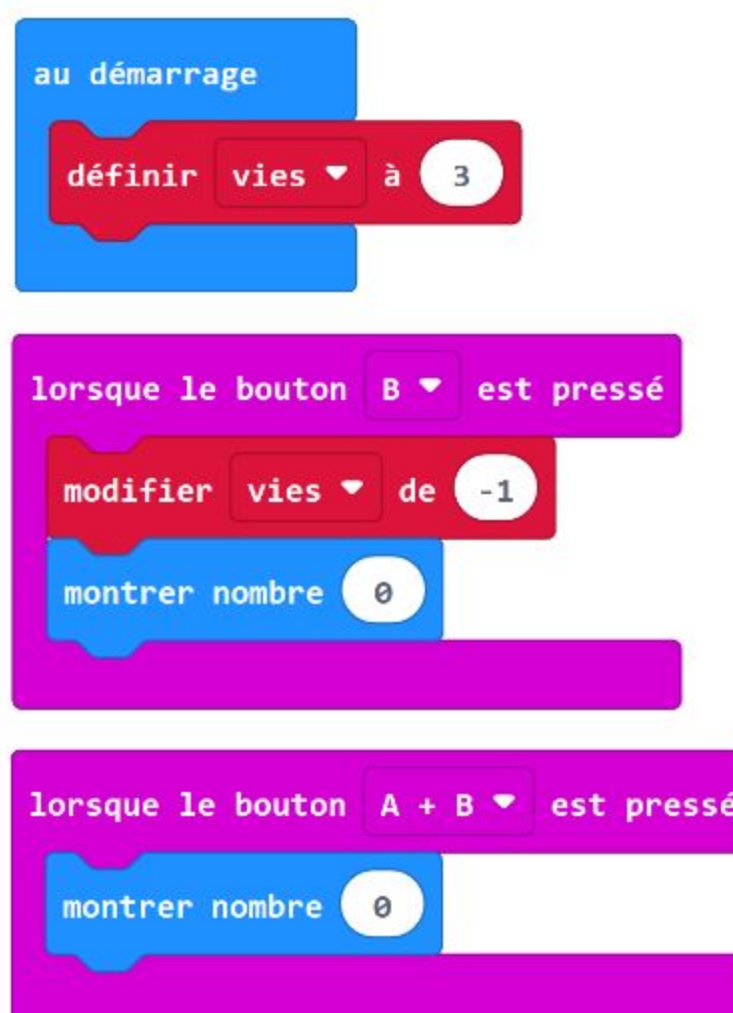
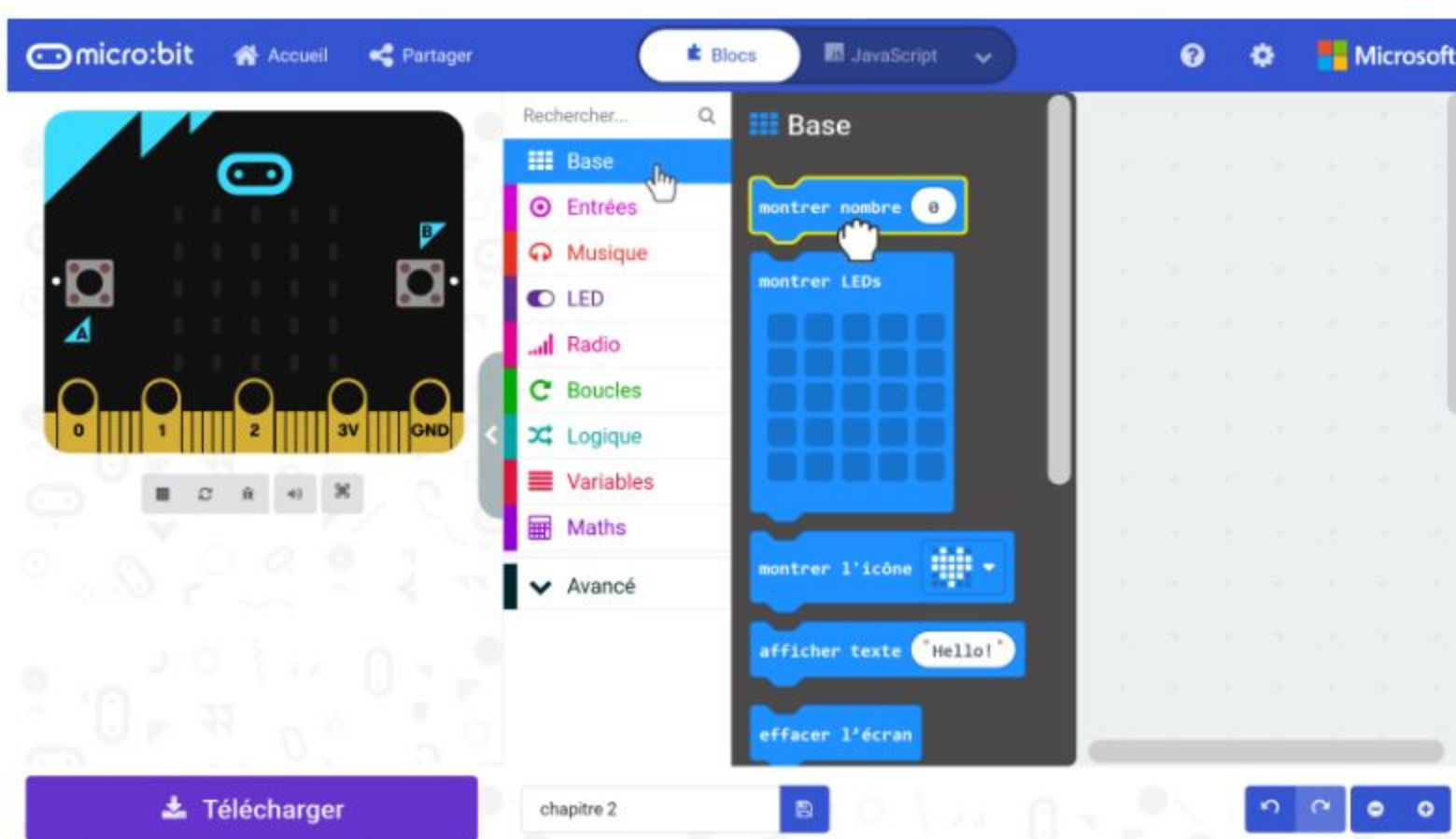




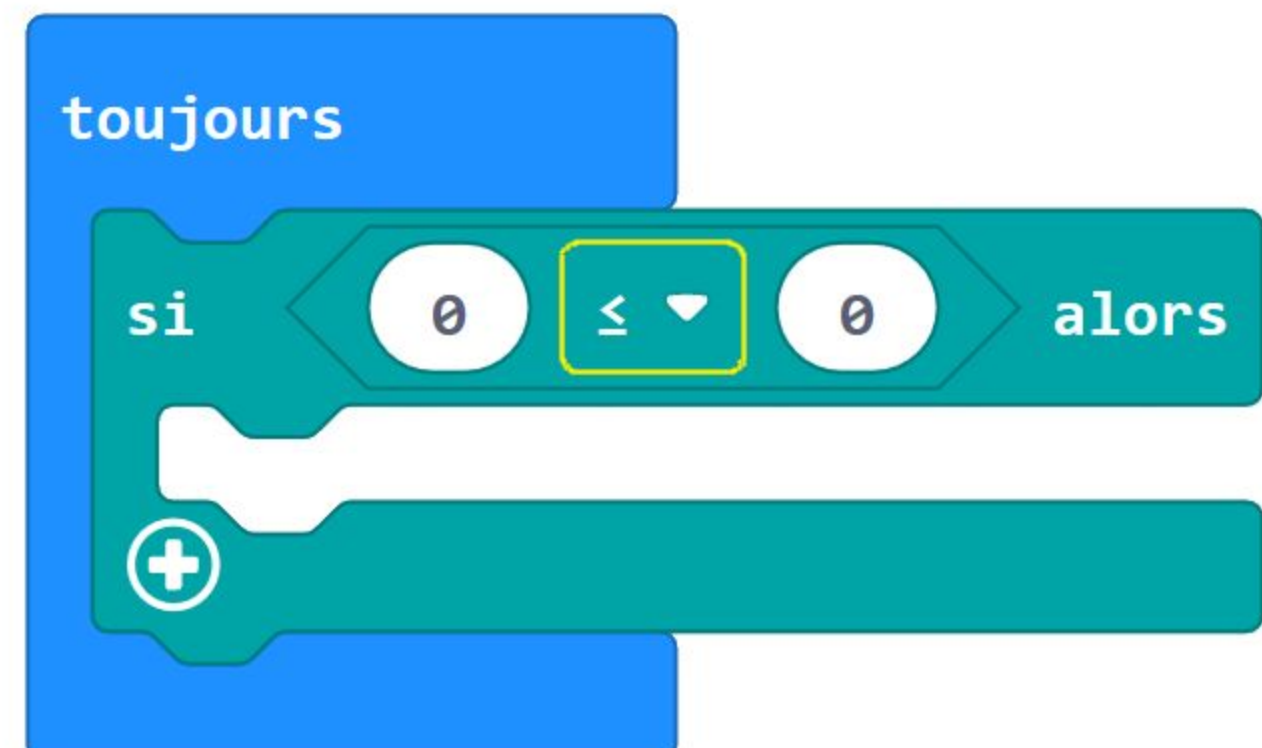
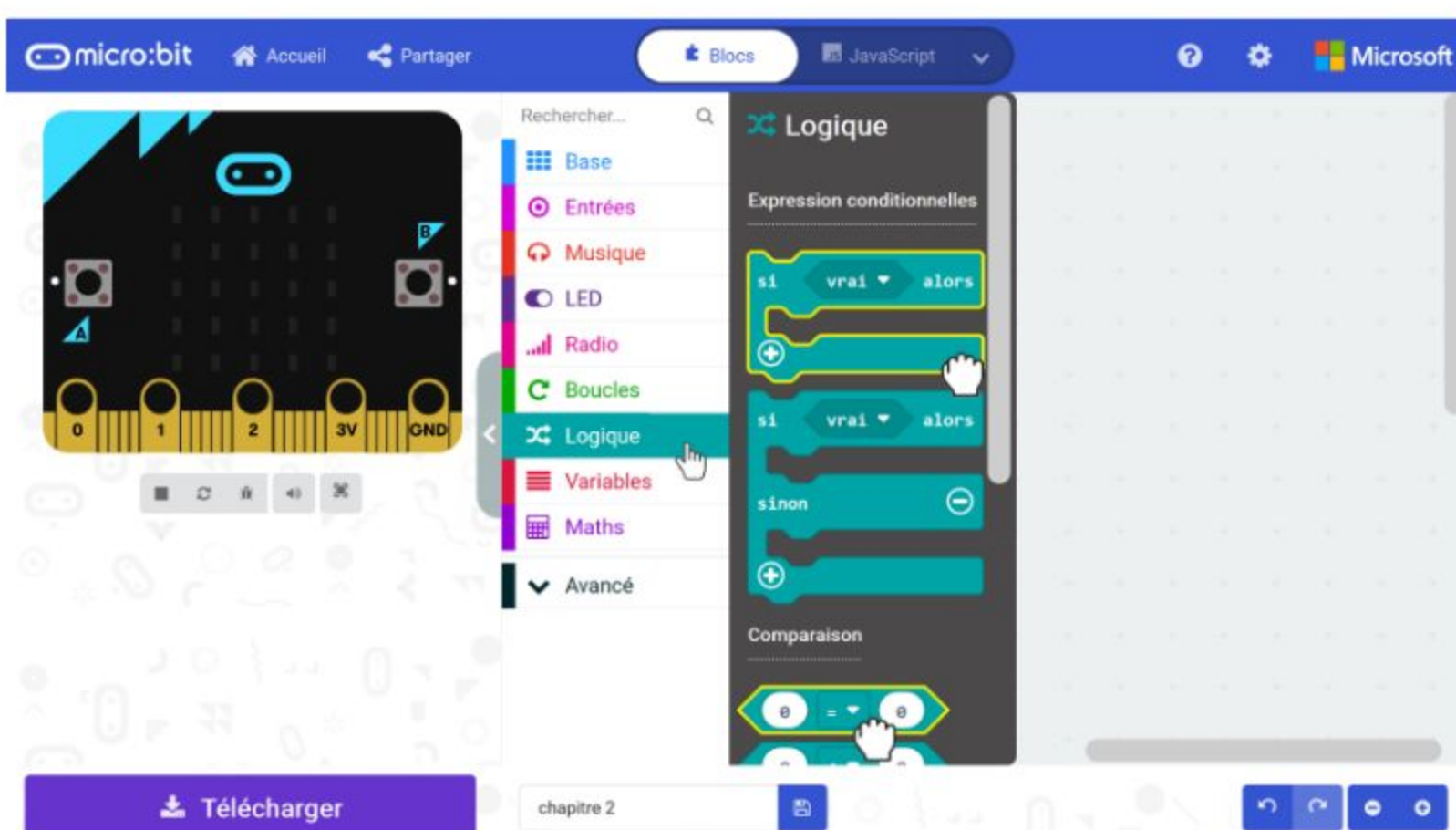
Étape 14 Clique à nouveau sur la catégorie **[Variables]** et sélectionne le bloc **[modifier _ de _]**. Imbrique-le dans le bloc **[lorsque le bouton B est pressé]**. Change la variable en “vies” et la valeur en -1.



Étape 15 Clique sur la catégorie **[Base]** et sélectionne le bloc **[montrer nombre]**. Duplique-le et imbrique-les dans les blocs **[lorsque le bouton _ est pressé]** B et A+B.

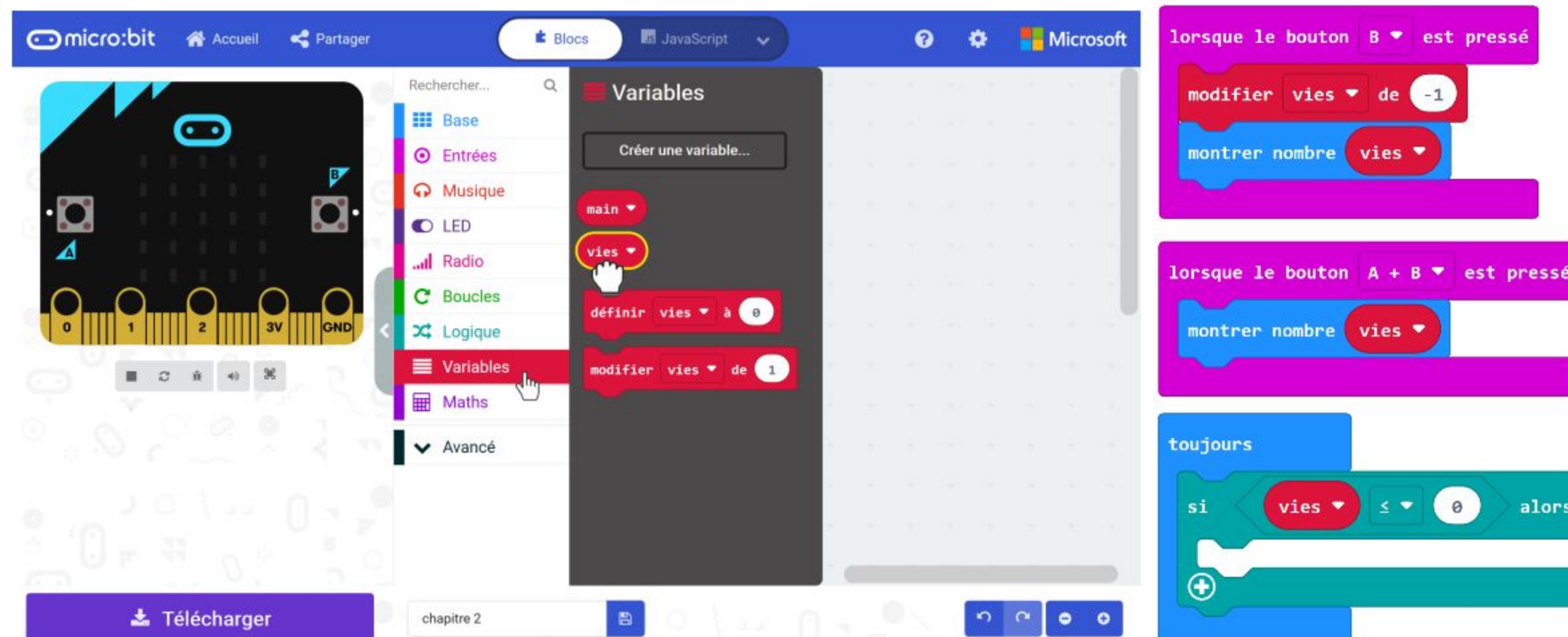


Étape 16 Clique sur la catégorie **[Logique]** et ajoute un bloc **[si-alors]**, puis insères-y un bloc de comparaison **[_ = _]** dans la condition. Imbrique-les dans le bloc **[Base]** : **[toujours]**. Change le signe “=” en “≤”.



CHAPITRE 2 : Jouons à Pierre, Papier, Ciseaux !

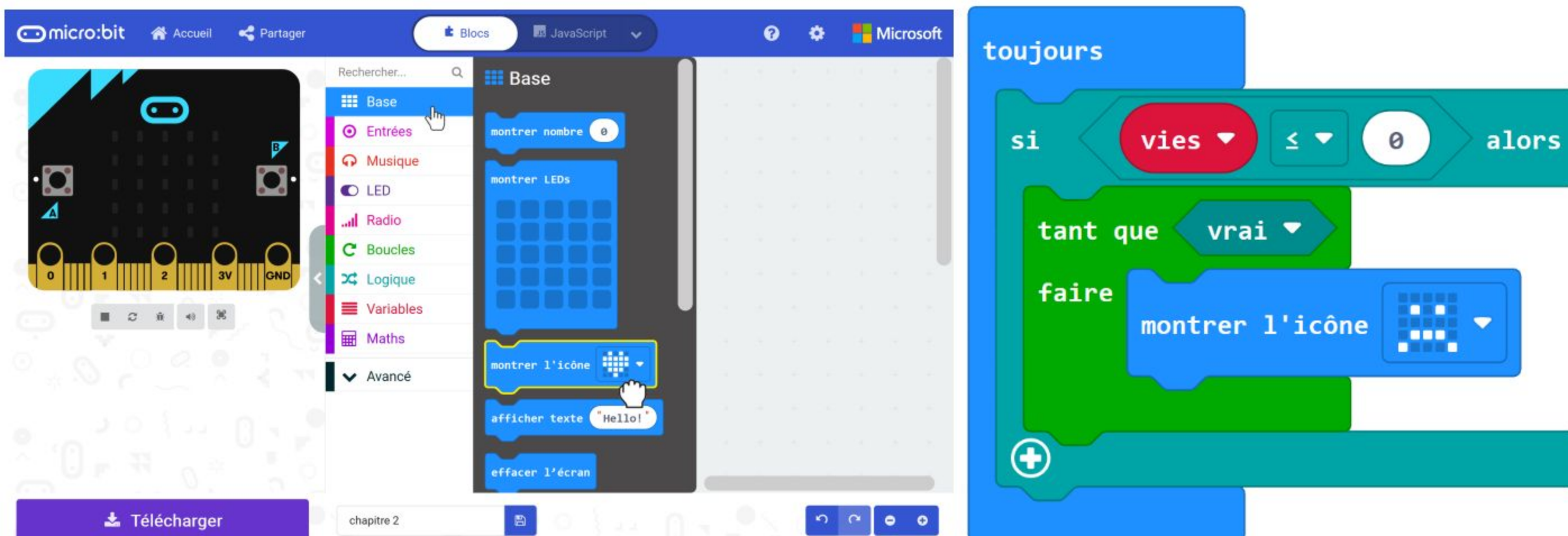
Étape 17 Clique sur la catégorie **[Variables]** et sélectionne le bloc **[vies]**. Duplique-le et imbrique-les dans les deux blocs **[montrer nombre]** et dans le membre de gauche du bloc de comparaison **[_ = _]**.



Étape 18 Clique sur la catégorie **[Boucles]** et sélectionne le bloc **[tant que _ faire]**. Imbrique-le dans le bloc **[si-alors]**.



Étape 19 Clique sur la catégorie **[Base]** et sélectionne le bloc **[montrer l'icône]**. Imbrique-le dans le bloc **[tant que _ faire]** et change l'icône en "triste".





Étape 20 Voici ci-dessous le code complet. Flashe-le sur EDU:BIT et amuse-toi avec tes amis pour déterminer le roi ou la reine du jeu Pierre Papier Ciseaux.

```

lorsque le bouton A est pressé
  définir main à choisir au hasard de 0 à 2
  si main = 0 alors
    montrer LEDs
  sinon si main = 1 alors
    montrer LEDs
  sinon
    montrer LEDs

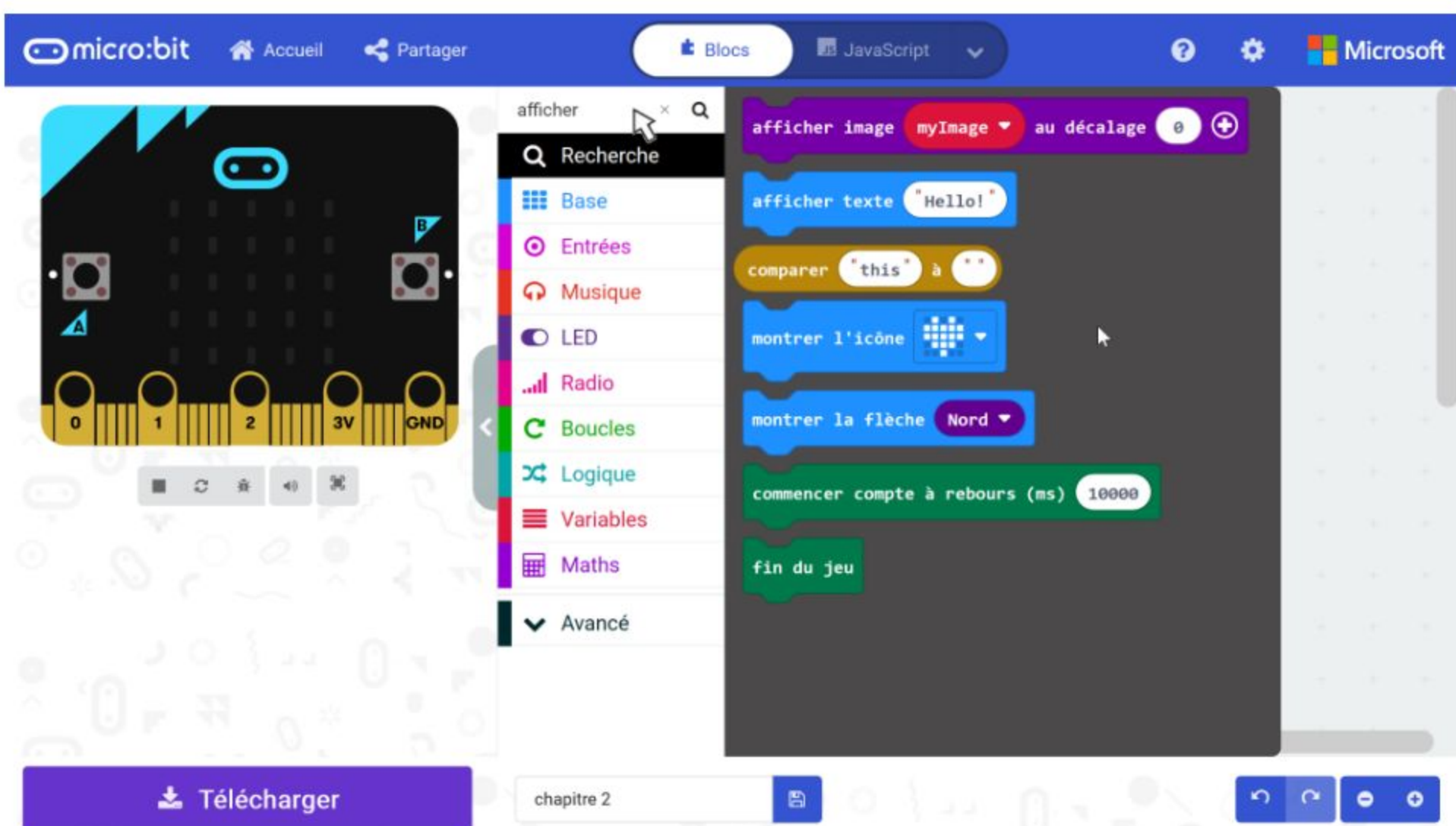
au démarrage
  définir vies à 3

lorsque le bouton B est pressé
  modifier vies de -1
  montrer nombre vies

lorsque le bouton A + B est pressé
  montrer nombre vies

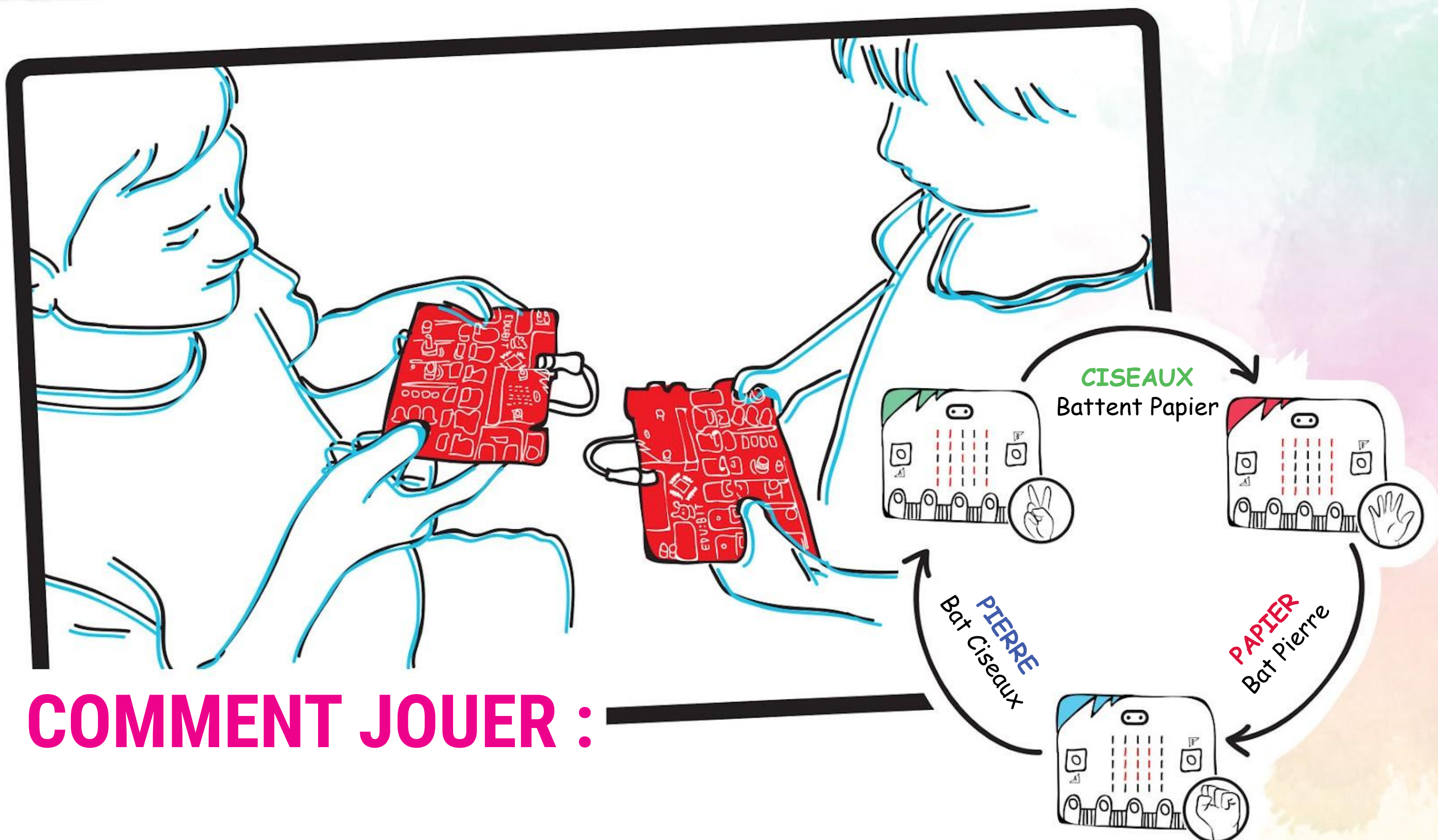
toujours
  si vies ≤ 0 alors
    tant que vrai
      faire
        montrer l'icône
  
```

Voici un conseil pour toi... Tous les blocs de codage ont un code couleur. Retrouve les blocs dont tu as besoin dans la catégorie dont le nom est de la même couleur. Tu peux aussi saisir des mots clés dans le champ "Rechercher..."



Jouons

Pierre Papier Ciseaux - Version améliorée



COMMENT JOUER :

- Les joueurs se placent face à face. Lorsqu'ils sont prêts, les deux joueurs appuient sur le bouton jaune (bouton A) pour afficher au hasard pierre, papier ou ciseaux.
- Comparer et désigner le vainqueur.
- Le perdant doit alors appuyer sur le bouton bleu (bouton B) de son EDU:BIT pour perdre une vie.
- Appuyez simultanément sur les deux boutons (jaune et bleu) pour vérifier et afficher le nombre de vies restant.
- Si un joueur a perdu 3 fois, la partie est finie et son EDU:BIT affichera un visage triste.

NOTE!



- Pour faire une nouvelle partie, tu dois relancer le programme.
- Si tu n'as personne avec qui jouer, tu peux toujours jouer contre le simulateur de l'éditeur MakeCode.

DÉCHIFFRE

LE CODE

En programmation informatique, on utilise des **variables** pour stocker des informations ou des valeurs qui peuvent évoluer au cours du temps (lorsque ton programme est en cours d'exécution). Tu peux te représenter une variable comme une enveloppe possédant un nom et contenant un morceau de papier sur lequel une information est écrite dessus. Ce morceau de papier peut être retiré et remplacé par un autre morceau de papier contenant une nouvelle information.



Dans notre code précédent, nous avons créé une variable nommée "vies" et lui avons affecté 3 comme valeur initiale.



L'information est le nombre 3 écrit sur le morceau de papier.

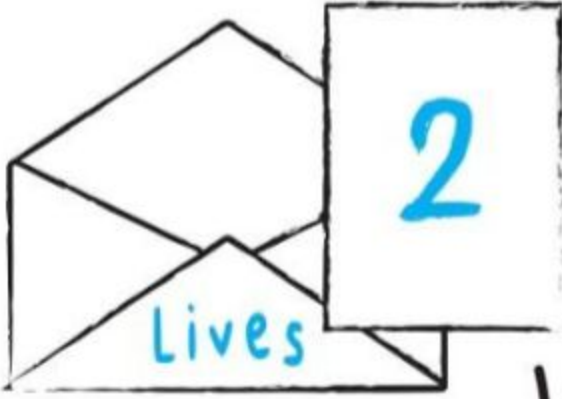

L'enveloppe nommée "vies" représente la variable.

Ensuite, lorsque le bouton B est pressé, la valeur est incrémentée de -1.

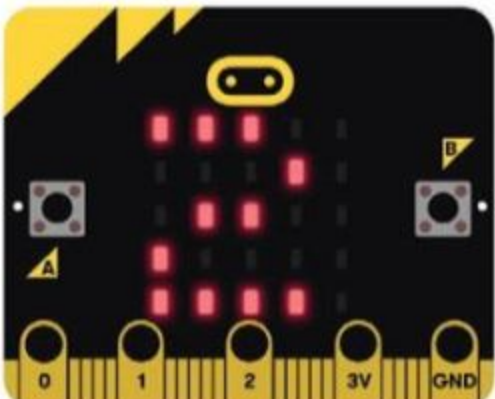


Lorsque le bouton B est pressé, le morceau de papier contenant l'information "3" est enlevé de l'enveloppe et remplacé par un autre morceau de papier avec la nouvelle information "2" (car : $3 - 1 = 2$).

Appuyer simultanément sur les boutons A+B affichera sur la matrice de LEDs la valeur actuelle de la variable "vies".



Lorsque les boutons A+B sont pressés, le morceau de papier est retiré de l'enveloppe et l'information dessus est lue. L'information retourne ensuite dans l'enveloppe.

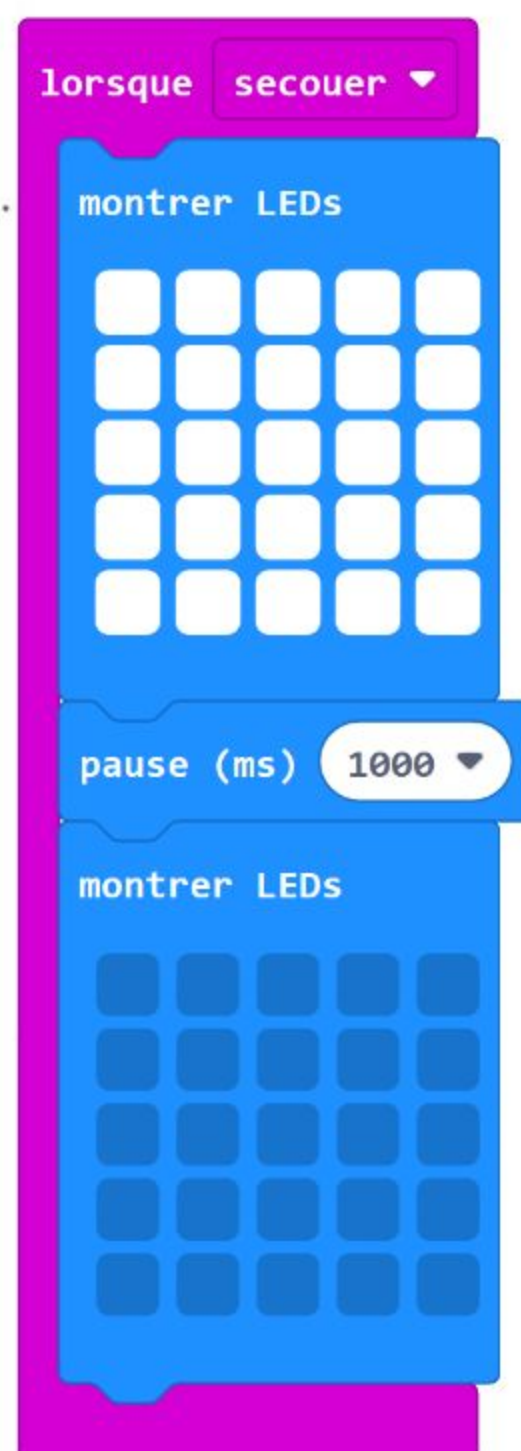




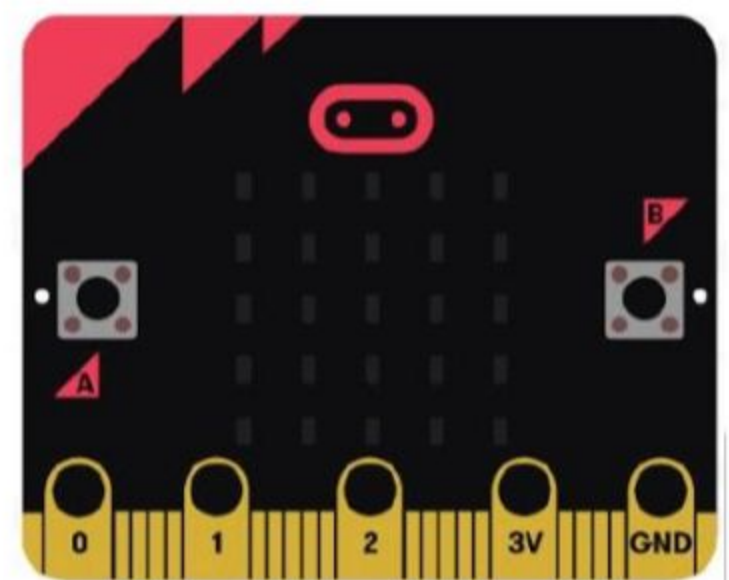
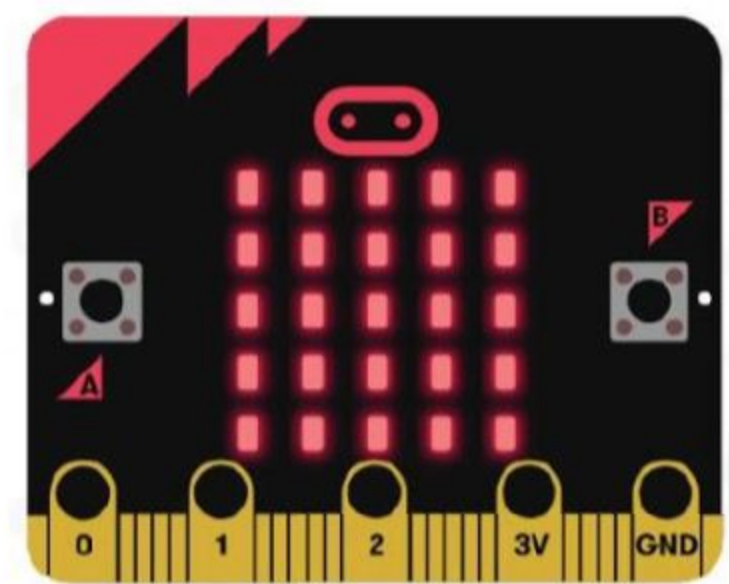
EXPLORE D'AUTRES BLOCS

En plus du bloc [**lorsque le bouton _ est pressé**], tu peux également utiliser d'autres blocs de la catégorie [**Entrées**] dans la programmation événementielle. Les actions réalisées par l'utilisateur comme appuyer sur un bouton ou secouer la carte s'appellent des événements.

Le code suivant, par exemple, allumera l'ensemble des LEDs de la matrice sur la carte pendant 1 seconde lorsque la carte est secouée. Essaie !



Résultat



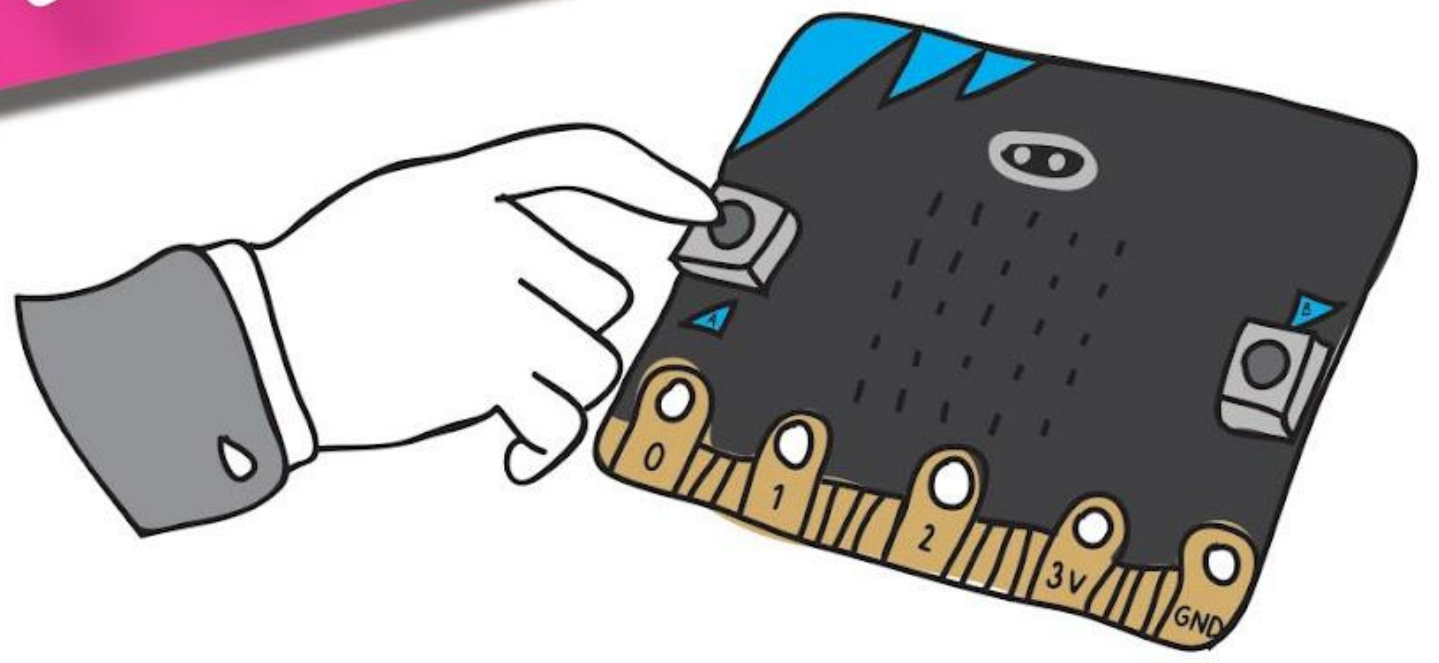
Si tu cliques sur le bouton [**secouer**] du bloc, un menu apparaît proposant une liste d'autres événements déclencheurs. Essaie de programmer EDU:BIT pour afficher une icône différente pour chacun d'entre-eux. Amuse-toi en les explorant !



EDU:BIT peut détecter les mouvements et connaît leur orientation grâce au capteur de mouvement intégré au micro:bit.

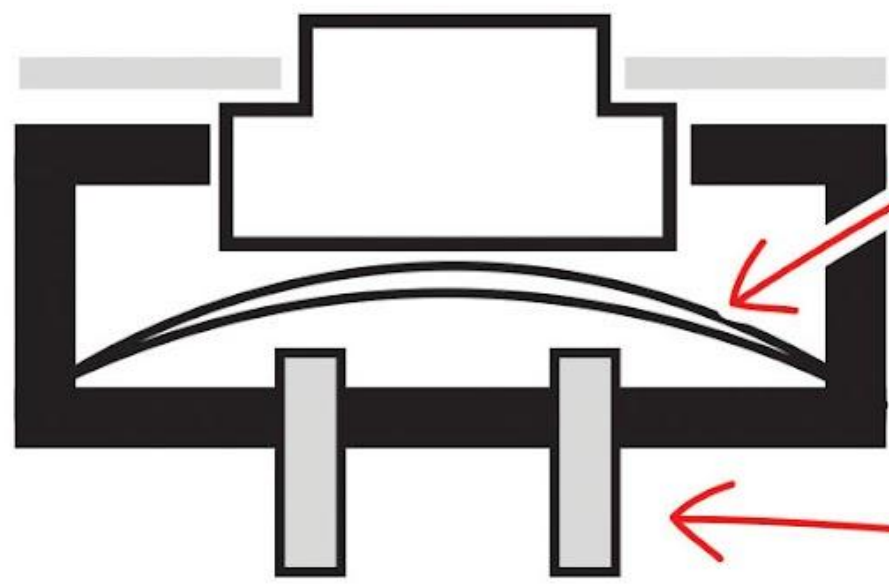


FAIT AMUSANT !

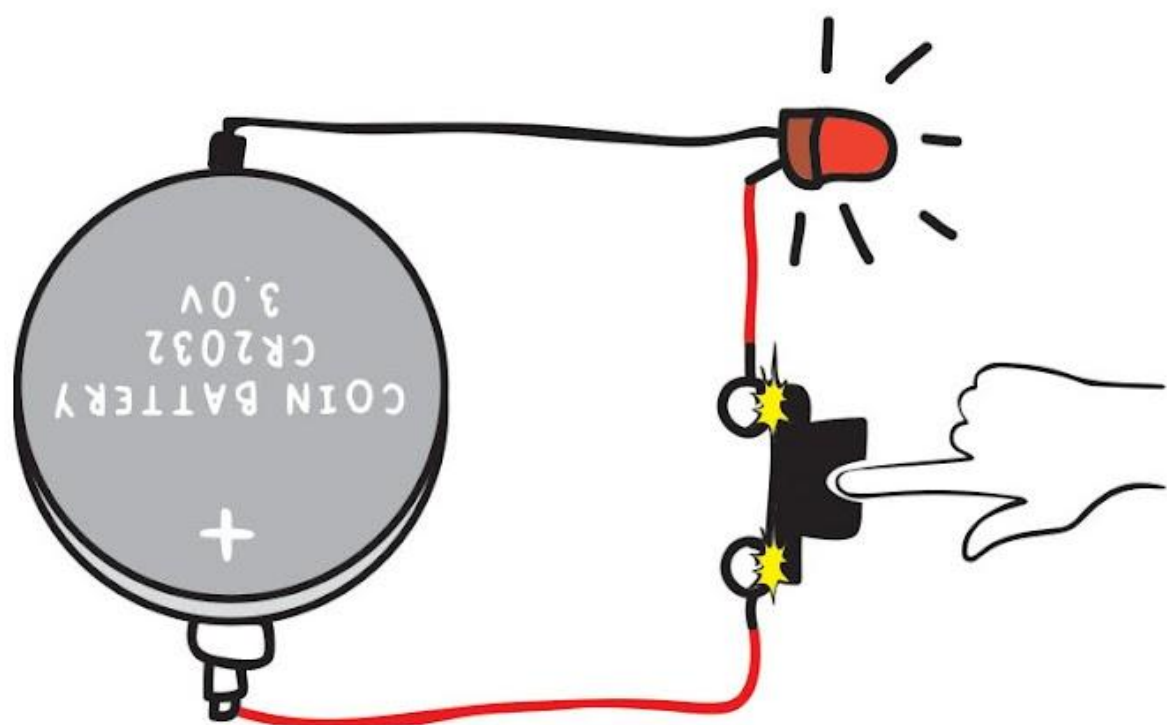
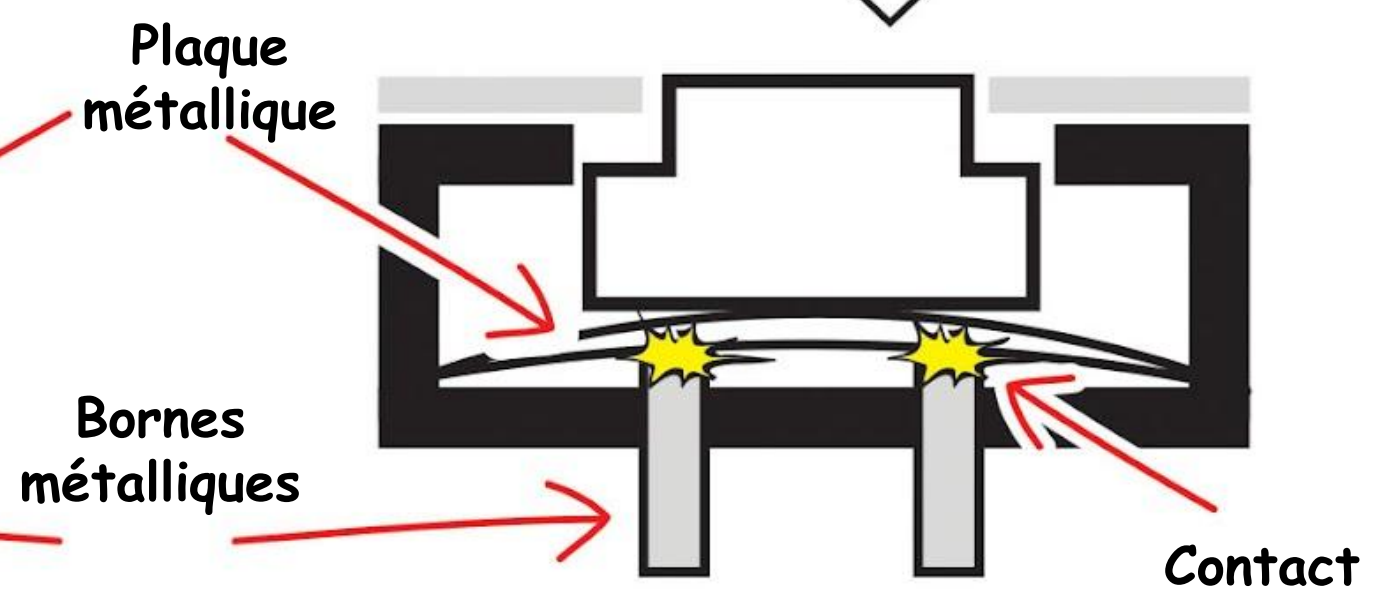


Un **bouton poussoir** est un périphérique d'entrée ayant seulement deux états : pressé et non pressé.

NON PRESSÉ



PRESSÉ



Lorsque le bouton poussoir est pressé, le circuit électrique est complété et la LED est allumée. Devine ce qui arrive lorsque l'on relâche le bouton ?



Les boutons noirs, gris, verts et blancs sont couramment utilisés pour la fonction MARCHÉ / ARRÊT, tandis que les boutons rouges sont utilisés pour les boutons paniques et les boutons d'urgence des machines.

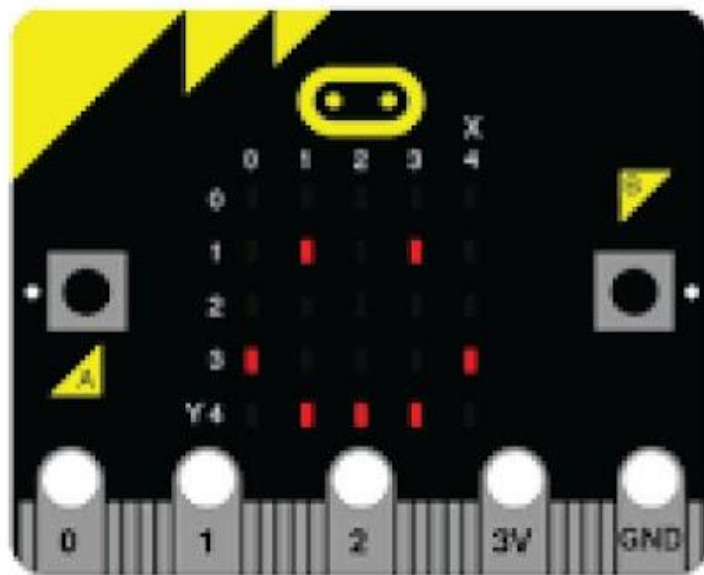


Apprends-en plus !



youtu.be/t_Qujjd_38o

APPLICATION DÉFI



Programme EDU:BIT pour qu'il serve de compteur d'élèves présents. Les filles doivent appuyer sur le bouton A lorsqu'elles entrent en classe, et les garçons doivent appuyer sur le bouton B.

au démarrage	Afficher une icône sourire. Initialiser deux variables nommées respectivement <i>filles</i> et <i>garçons</i> à 0.
lorsque le bouton A est pressé (jaune)	Incrémenter la variable <i>filles</i> de 1.
lorsque le bouton B est pressé (bleu)	Incrémenter la variable <i>garçons</i> de 1.
lorsque les boutons A+B sont pressés	Afficher les informations suivantes sur l'écran d'affichage : Total = (<i>filles</i> + <i>garçons</i>) ; Filles = (<i>filles</i>) ; Garçons = (<i>garçons</i>).

CHAPITRE 3

Un peu de Musique~

Music Bit (buzzer piézoélectrique + jack audio)



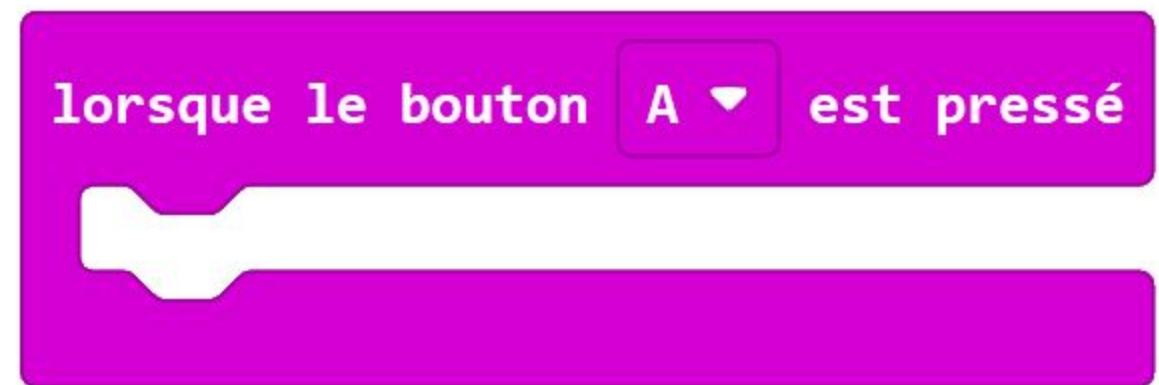
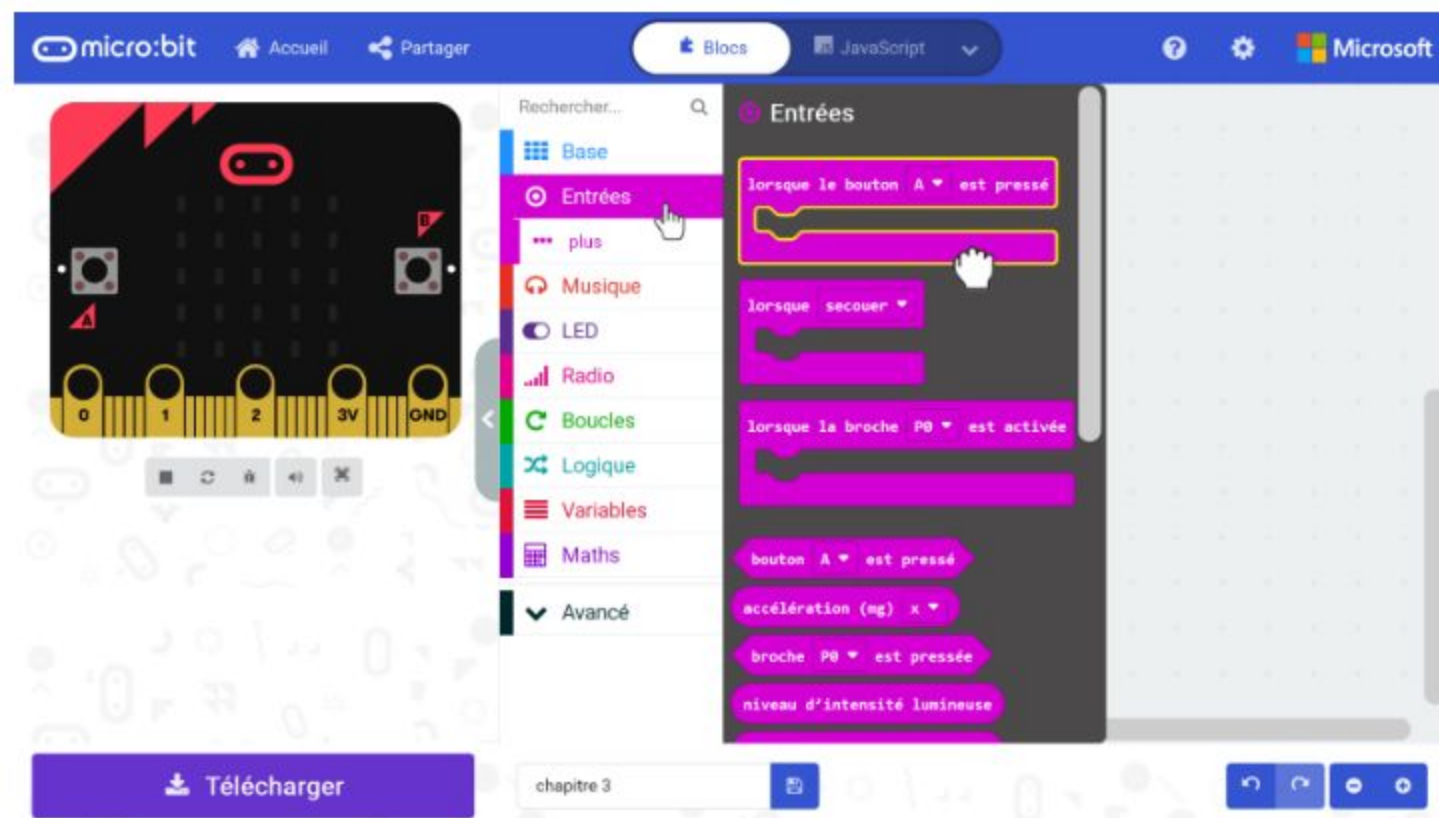
Scanne-moi !



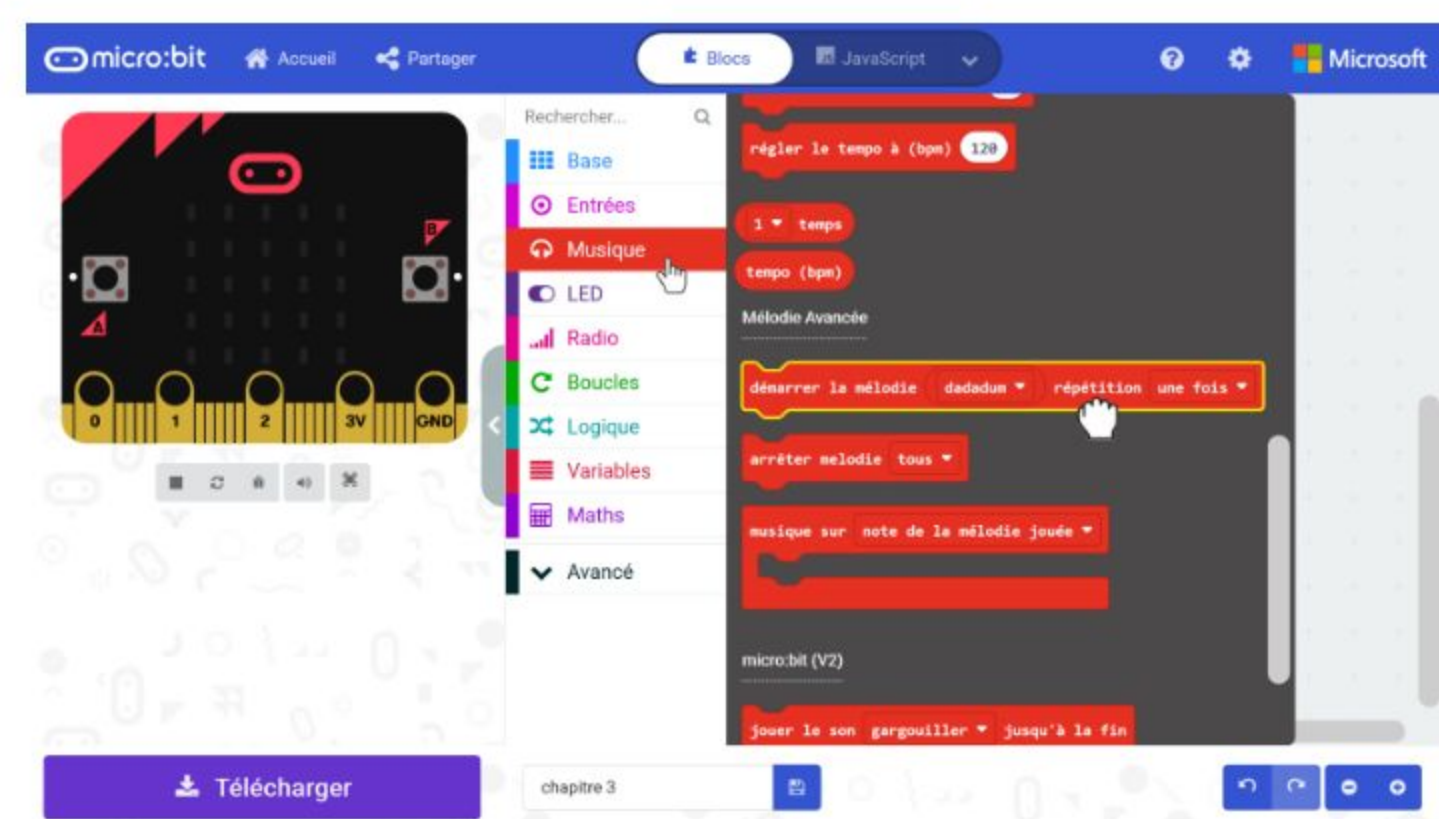
link.cytron.io/edubit-chapter-3

CODONS !

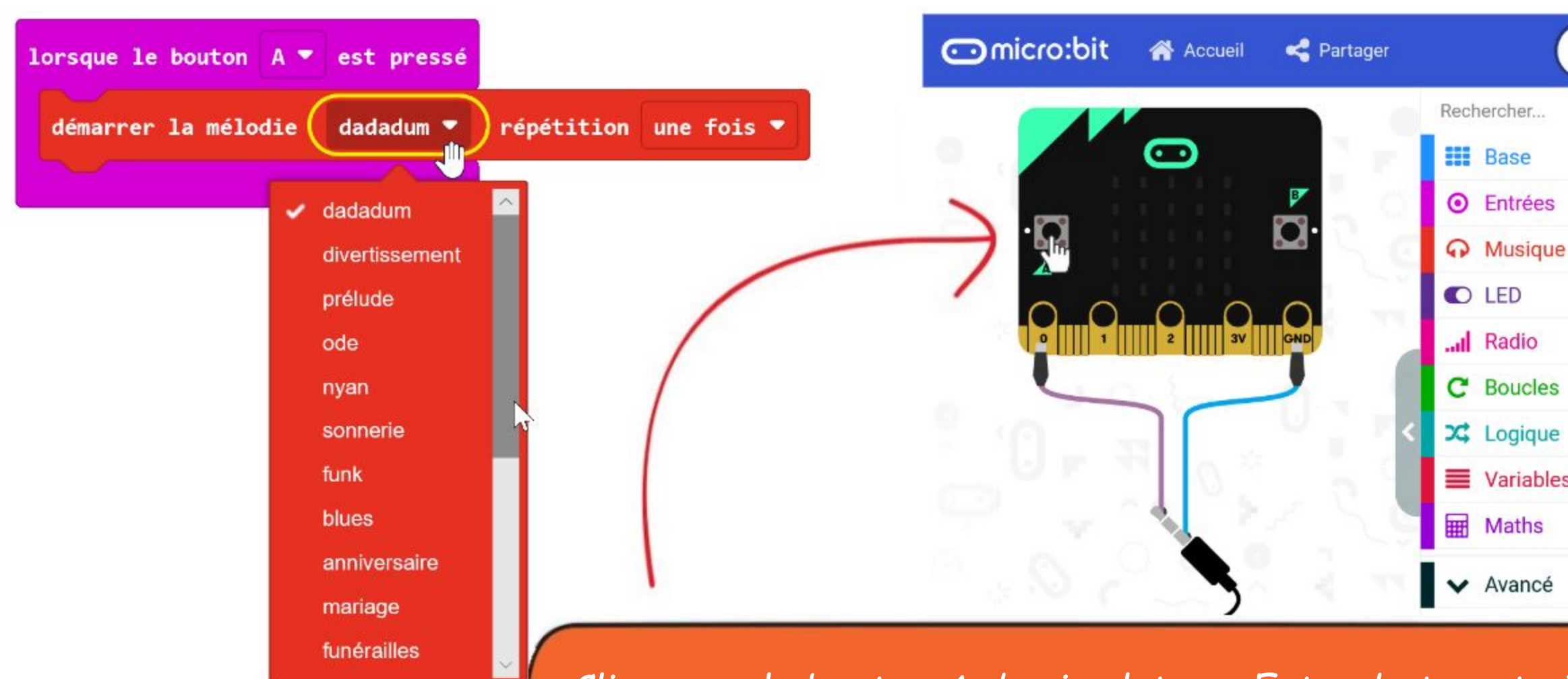
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode. Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**].



Étape 2 Clique sur la catégorie [**Musique**] et sélectionne le bloc [**démarrer la mélodie _ répétition _**].



Étape 3 Clique sur le bouton [**dadadum**] et choisis "anniversaire" dans la liste déroulante.



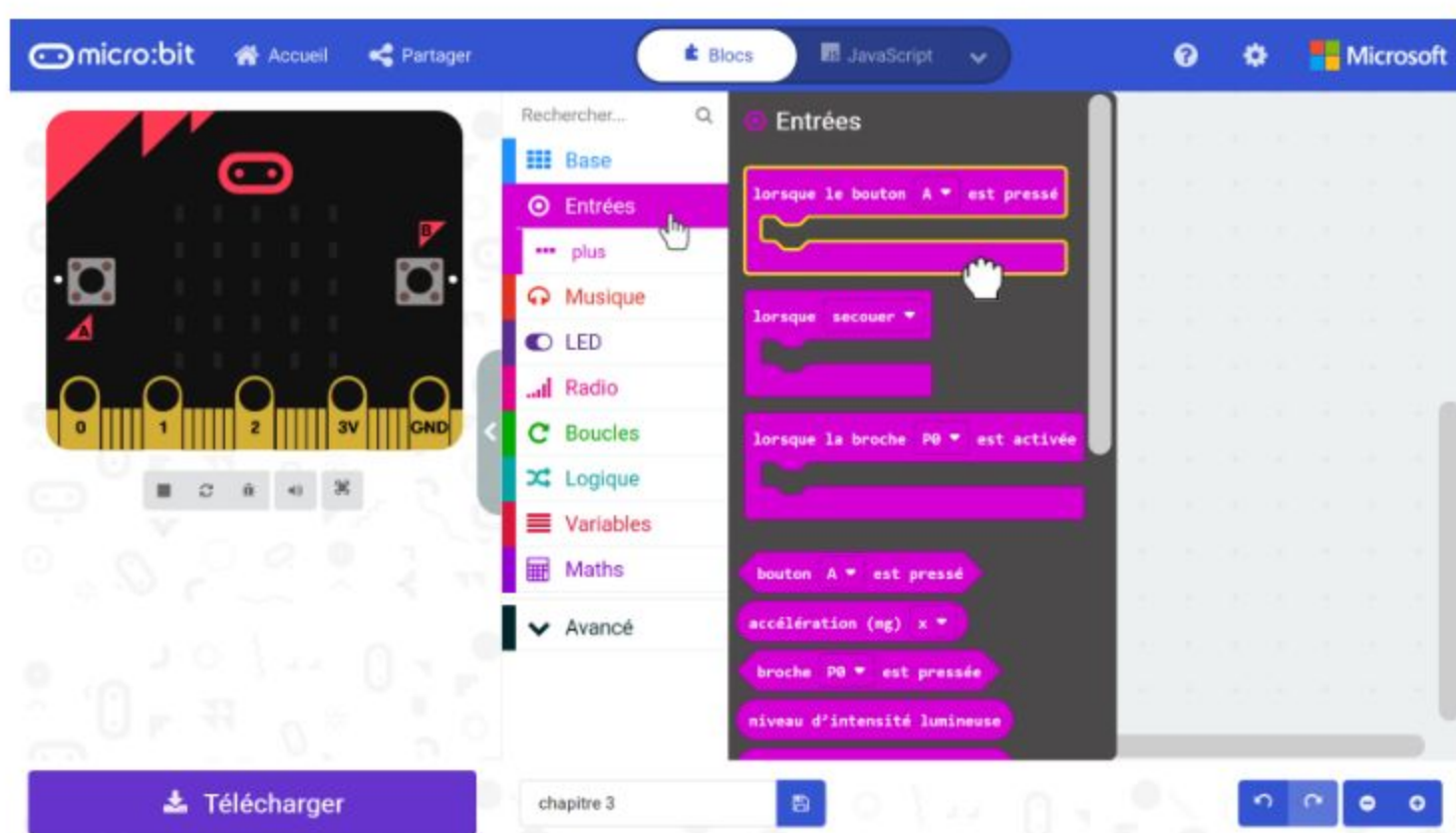
*Clique sur le bouton A du simulateur. Entends-tu cet air familier ? Amuse-toi à écouter les autres mélodies de la liste~ *assure-toi que les haut-parleurs de ton ordinateur sont allumés.*



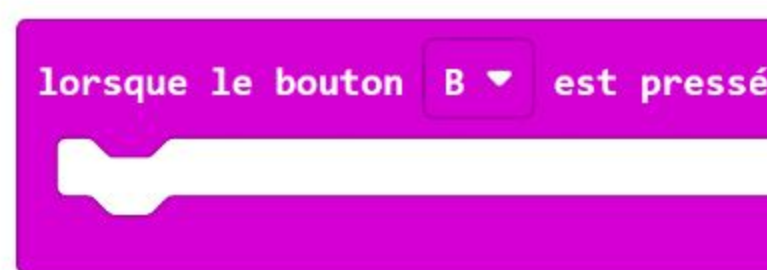
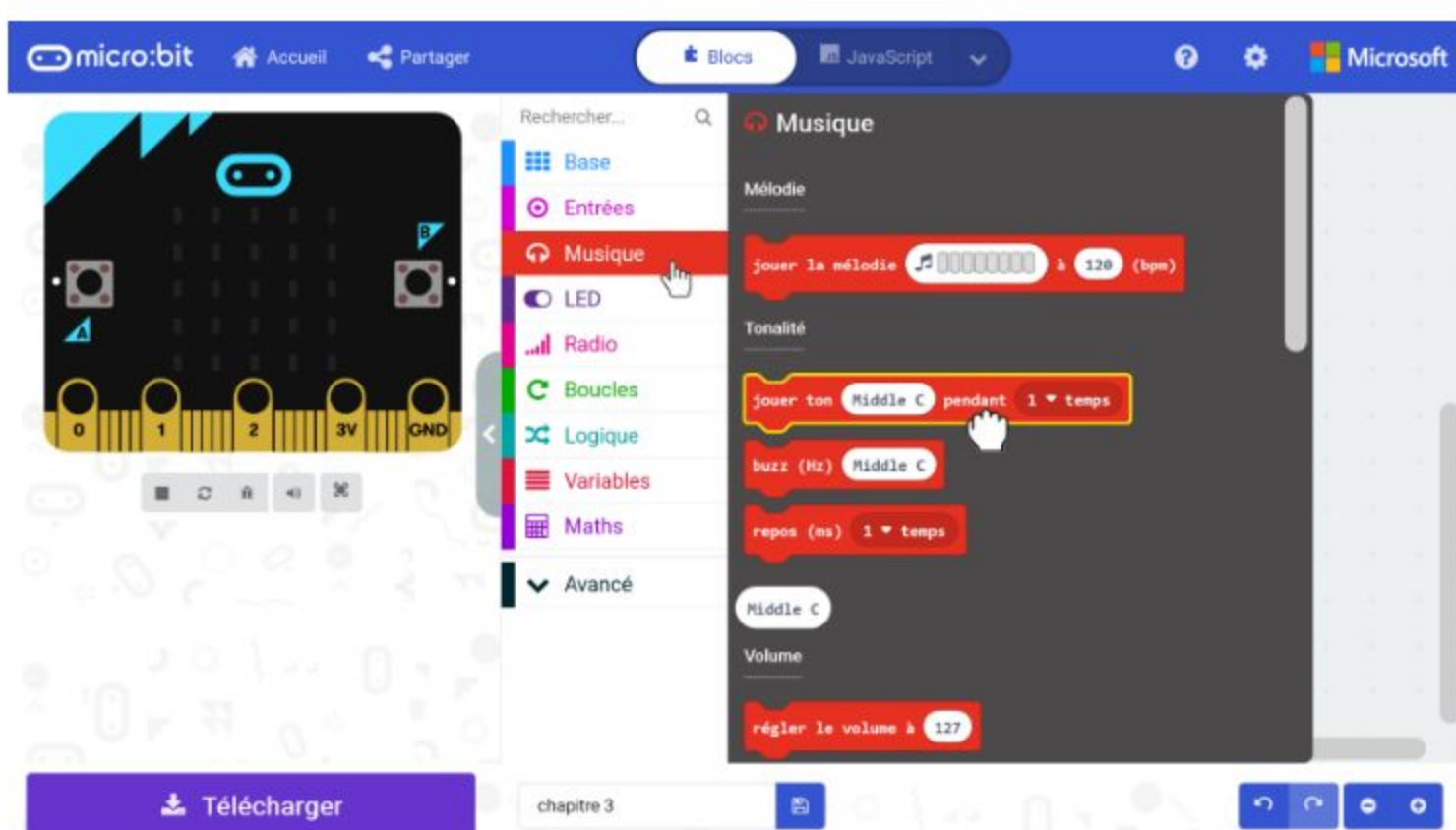
En plus de la liste de mélodies prédéfinies, tu peux programmer EDU:BIT pour jouer n'importe quelle chanson que tu aimes. Essayons de jouer cet air entraînant~



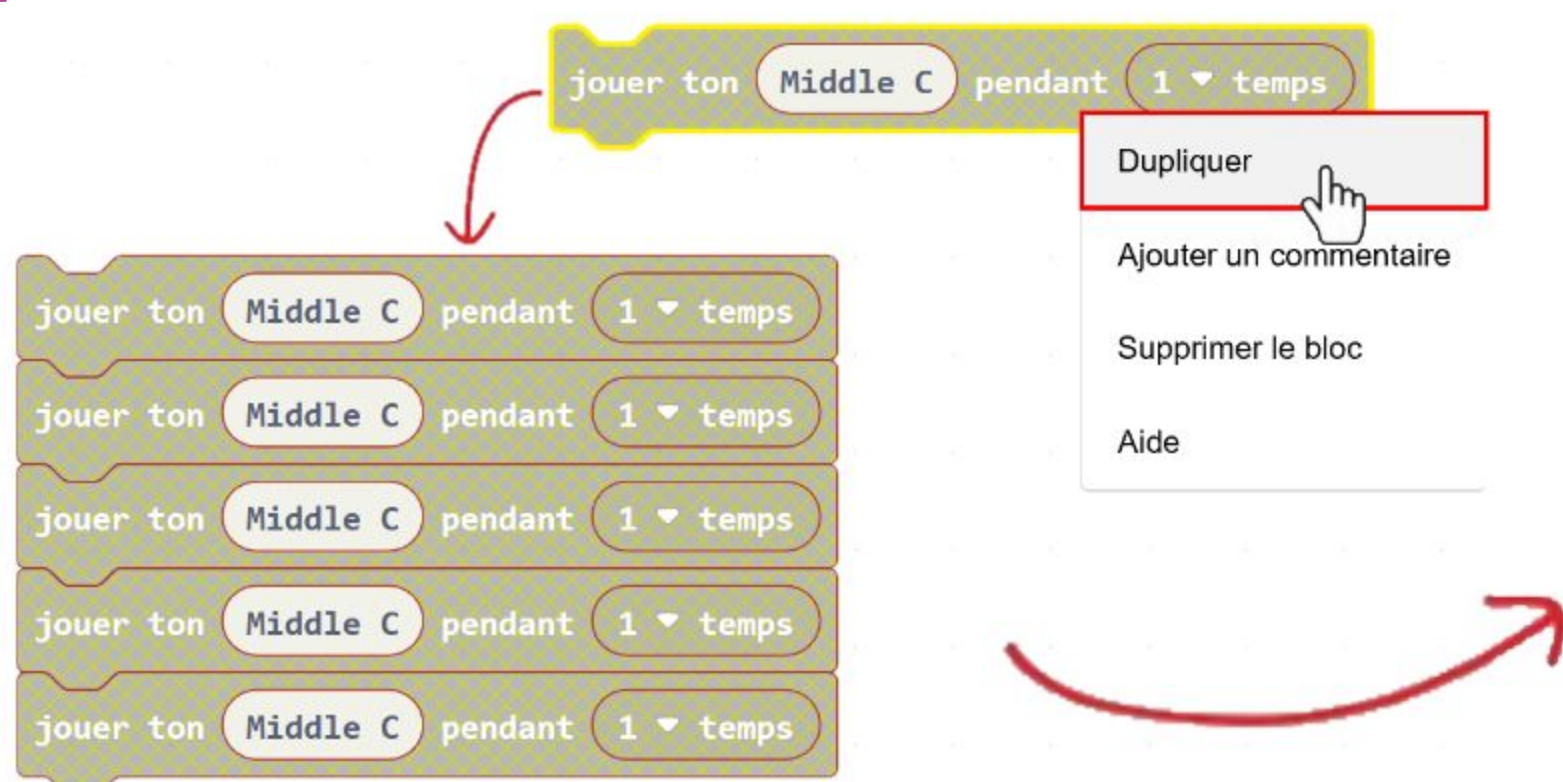
Étape 4 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**]. Change le bouton en "B".



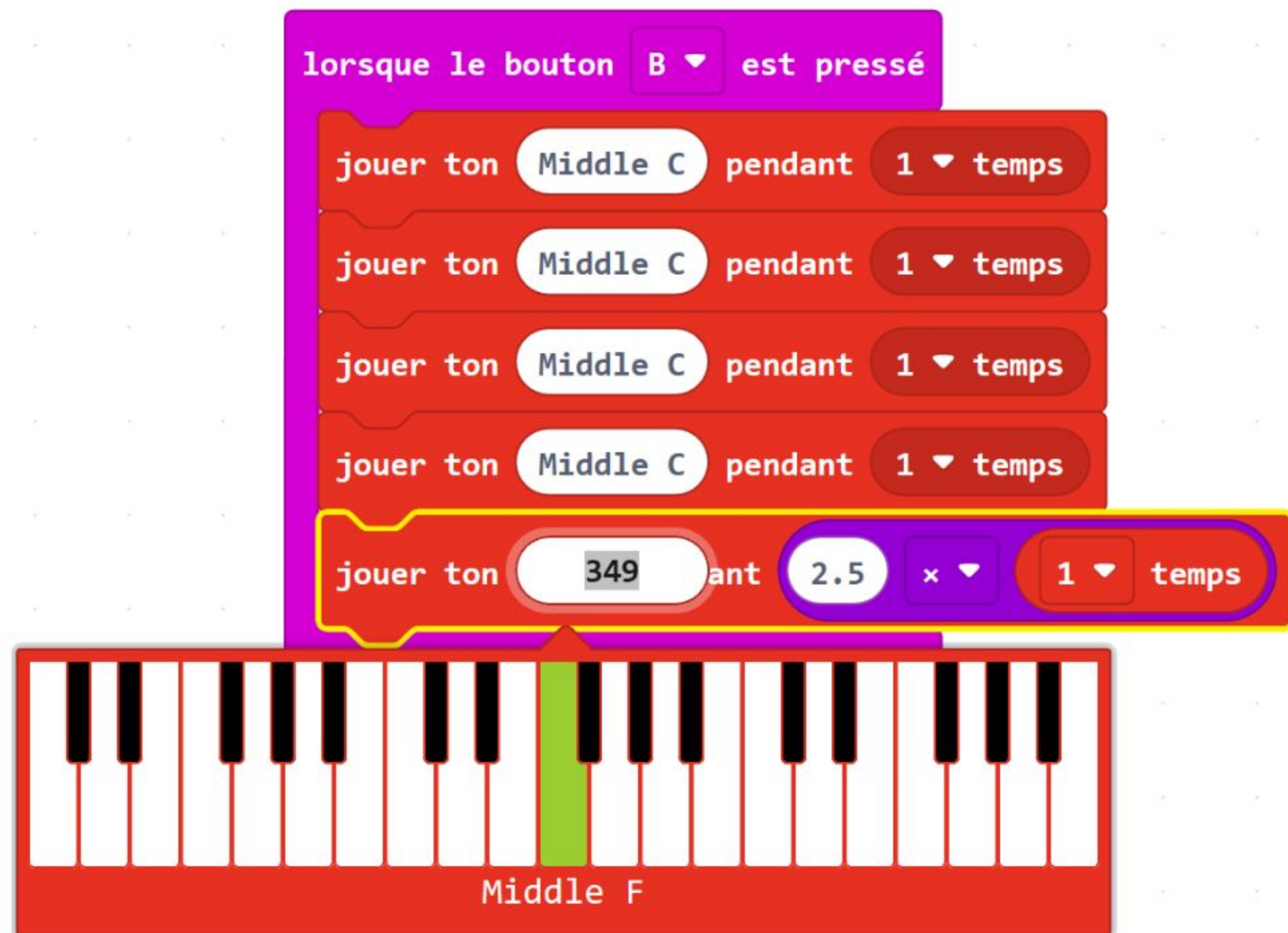
Étape 5 Clique sur [**Musique**] et sélectionne le bloc [**jouer ton _ pendant _ temps**].



Étape 6 Dans l'espace de travail, fais un clic droit sur le bloc [**jouer ton _ pendant _ temps**] et clique sur "Dupliquer". Répète l'opération jusqu'à obtenir 5 blocs [**jouer ton _ pendant _ temps**] en tout. Imbrique-les dans le bloc [**lorsque le bouton B est pressé**].

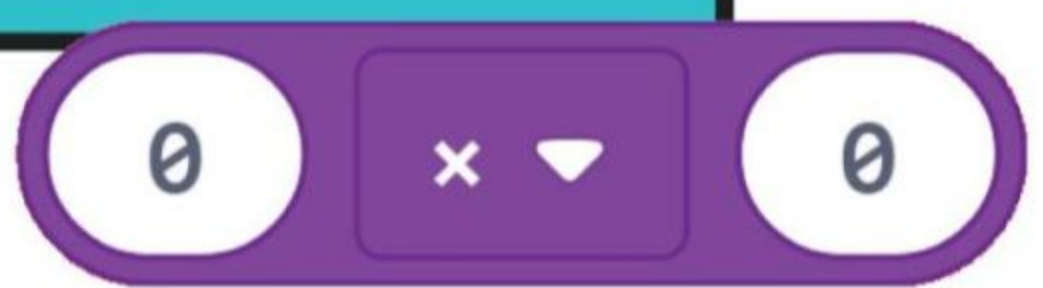


Étape 7 Pour chacun des blocs [**jouer ton _ pendant _ temps**], définis la note ("ton") et sa durée ("temps"), en suivant l'exemple ci-dessous.



NOTE!

Le bloc violet provient de la catégorie [Maths].



*Clique sur le bouton B du simulateur.
Peux-tu deviner la chanson ?*



Lorsque tu programmes, il est conseillé de vérifier régulièrement ton code pour être certain d'être sur la bonne voie. Tu peux utiliser le simulateur pour cela.

Étape 8 Code le reste de la chanson en ajoutant d'autres blocs [**jouer ton _ pendant _ temps**] ainsi qu'en choisissant les notes et les durées correspondantes. Tu peux te référer à la page suivante pour les connaître.



I Will Follow You

I will fol-low you, fol-low you wher-ev-er you may go, There is-n't an o-cean too

deep, a moun-tain so high it can keep keep me a - way

lorsque le bouton B est pressé

I will follow you,

- jouer ton Middle C pendant 1/2 temps
- jouer ton Middle D pendant 1/2 temps
- jouer ton Middle F pendant 1/2 temps
- jouer ton Middle G pendant 1/2 temps
- jouer ton Middle F pendant 2.5 x 1 temps
- repos (ms) 1/2 temps

Follow you wherever you may go,

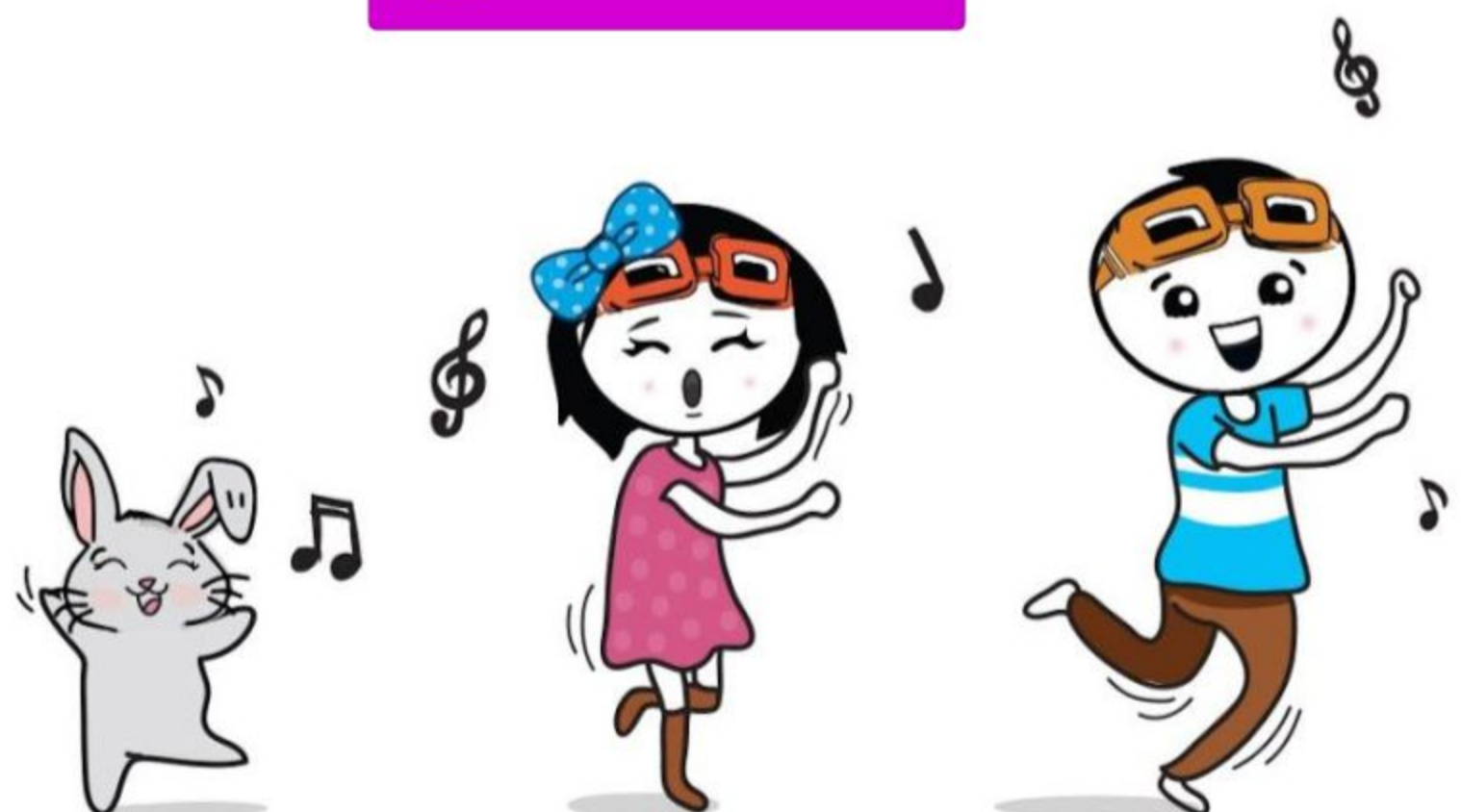
- jouer ton Middle C pendant 1/2 temps
- jouer ton Middle D pendant 1/2 temps
- jouer ton Middle F pendant 1/2 temps
- jouer ton Middle D pendant 1/2 temps
- jouer ton Middle F pendant 1/2 temps
- jouer ton Middle A pendant 1/2 temps
- jouer ton High C pendant 1.5 x 1 temps
- jouer ton Middle A pendant 1/2 temps
- jouer ton High C pendant 2 temps
- repos (ms) 1/2 temps

There is n't an ocean too deep

- jouer ton High C pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton Middle A pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High C pendant 2 temps
- repos (ms) 1/2 temps

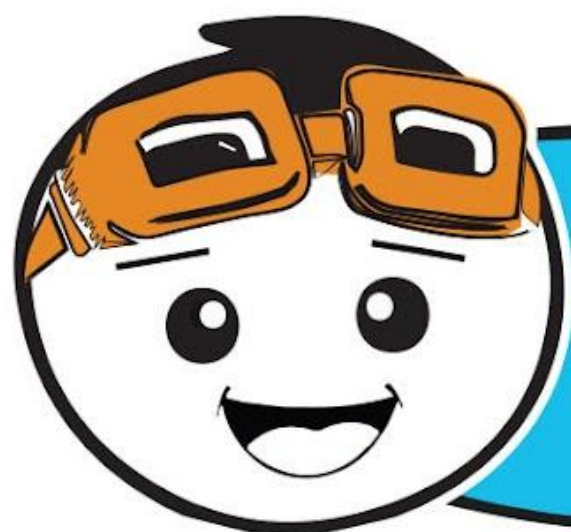
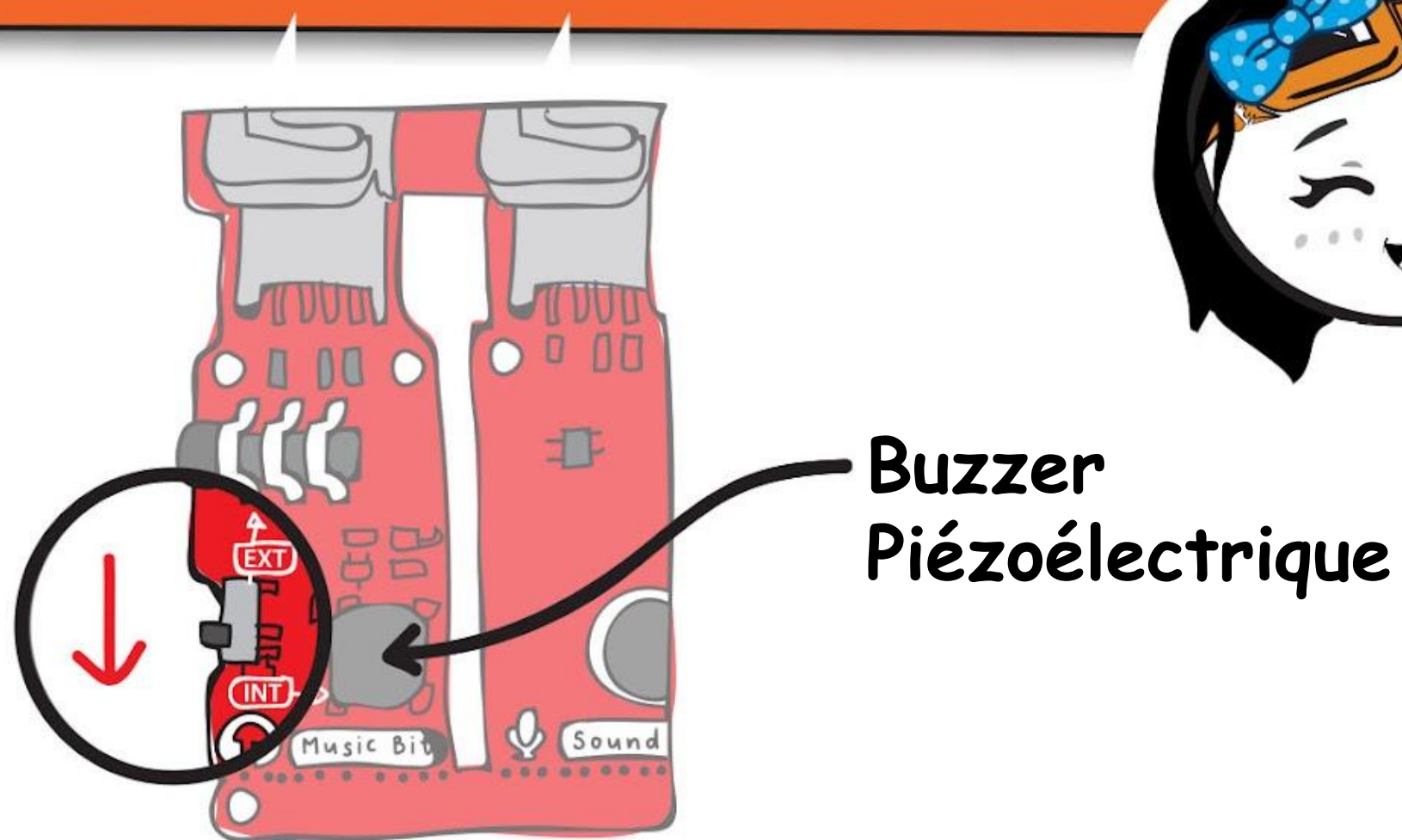
A mountain so high it can keep, Keep me a-way

- jouer ton High C pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High D pendant 1/2 temps
- jouer ton High C pendant 1/2 temps
- jouer ton Middle B pendant 1/2 temps
- jouer ton High C pendant 2 temps
- repos (ms) 1/2 temps
- jouer ton High C pendant 1/2 temps
- jouer ton Middle A pendant 1 temps
- jouer ton Middle G pendant 1 temps
- jouer ton Middle A pendant 1 temps
- jouer ton Middle G pendant 1/2 temps
- jouer ton Middle F pendant 2.5 x 1 temps
- repos (ms) 1 temps



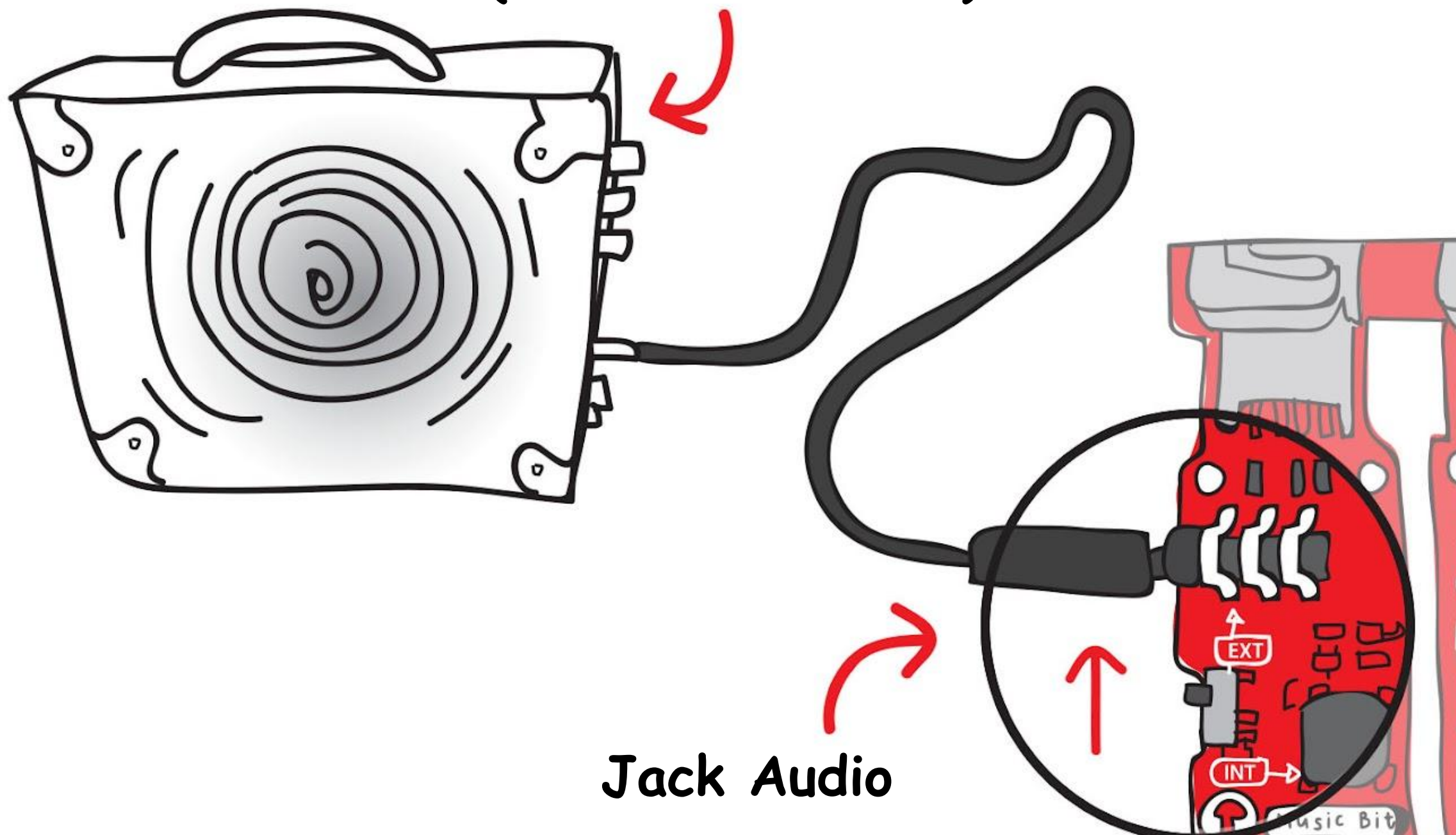
Étape 9 Flashe le code complet sur EDU:BIT.

EDU:BIT jouera la chanson "I Will Follow You" lorsque tu appuieras sur le bouton bleu (bouton B) de la carte. Pense à allumer EDU:BIT et le buzzer piézoélectrique en positionnant l'interrupteur coulissant sur INT (interne).



Tu peux également brancher des enceintes externes, des haut-parleurs ou un casque audio sur la prise jack d'EDU:BIT. Tu dois positionner l'interrupteur à coulisse sur EXT (externe).

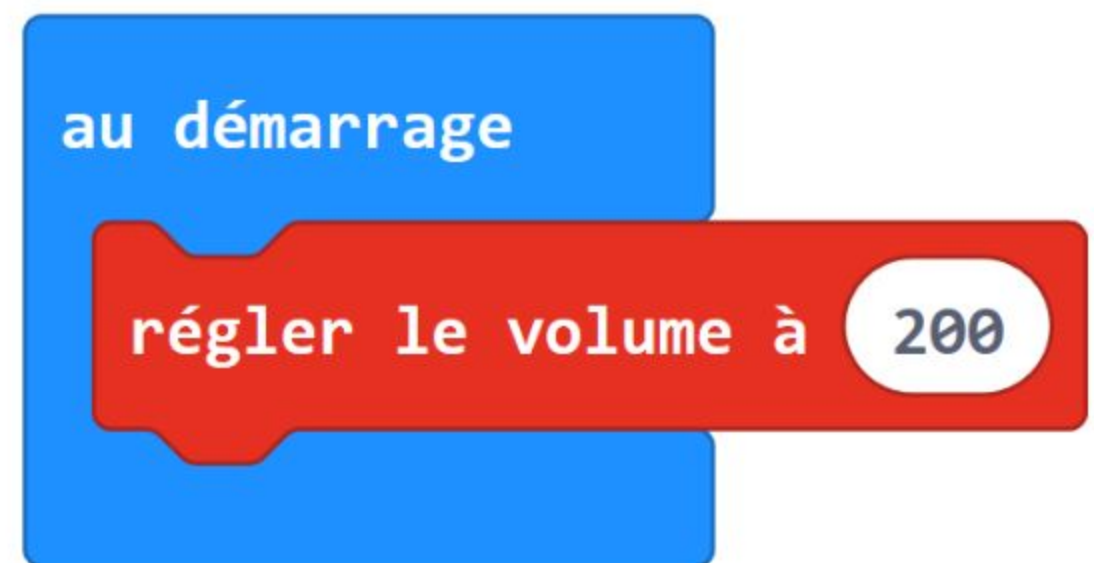
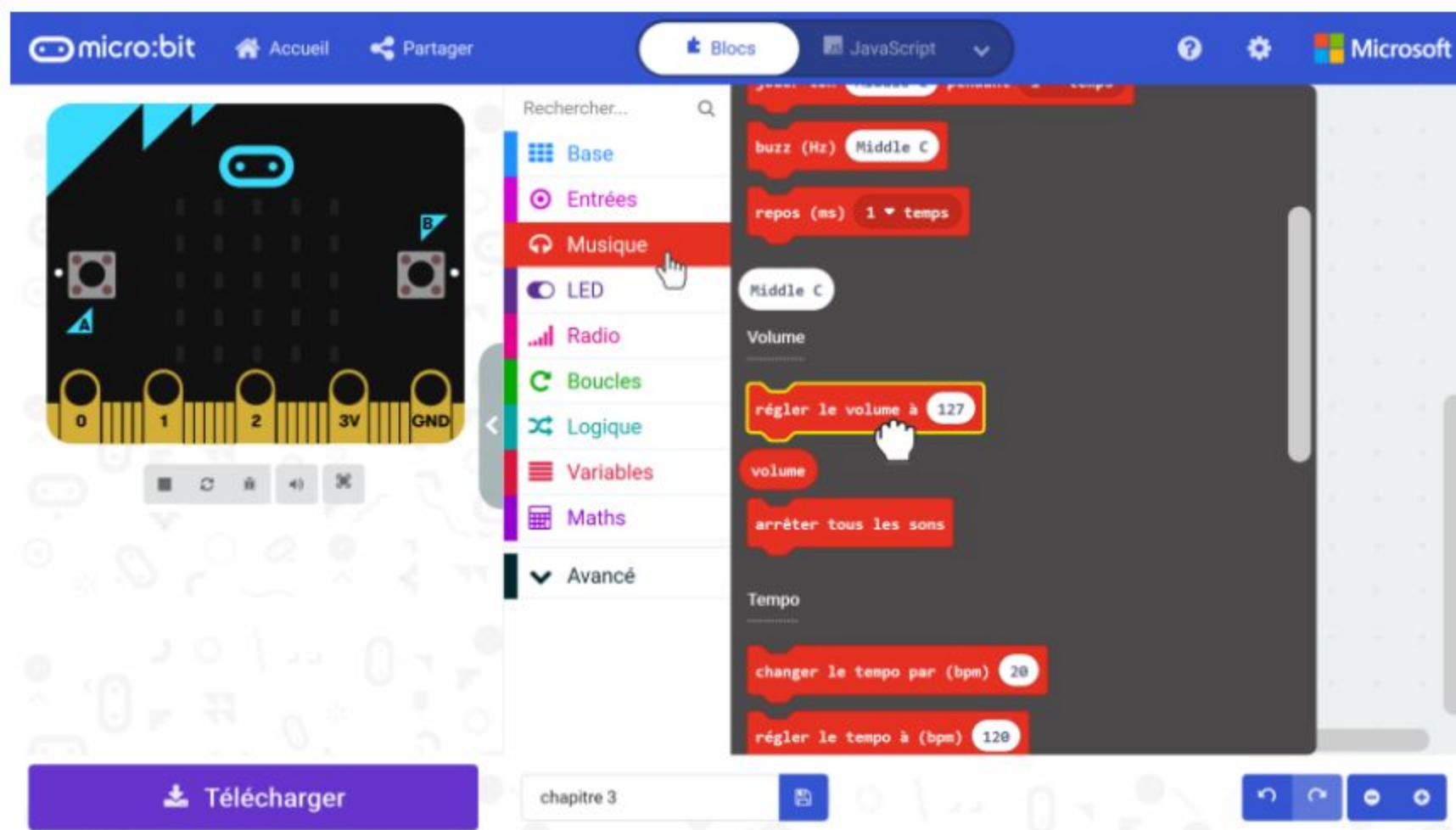
**Enceinte
(ou Haut-Parleurs)**





Le son est trop faible ? Ou trop fort ? Tu peux ajouter un bloc [régler le volume à _] pour changer le volume, qui prend des valeurs comprises entre 0 et 255 (volume maximal).

Étape 10 Clique sur la catégorie [**Musique**] et sélectionne le bloc [**régler le volume à _**]. Imbrique-le dans le bloc [**au démarrage**] et remplace la valeur par **200**.

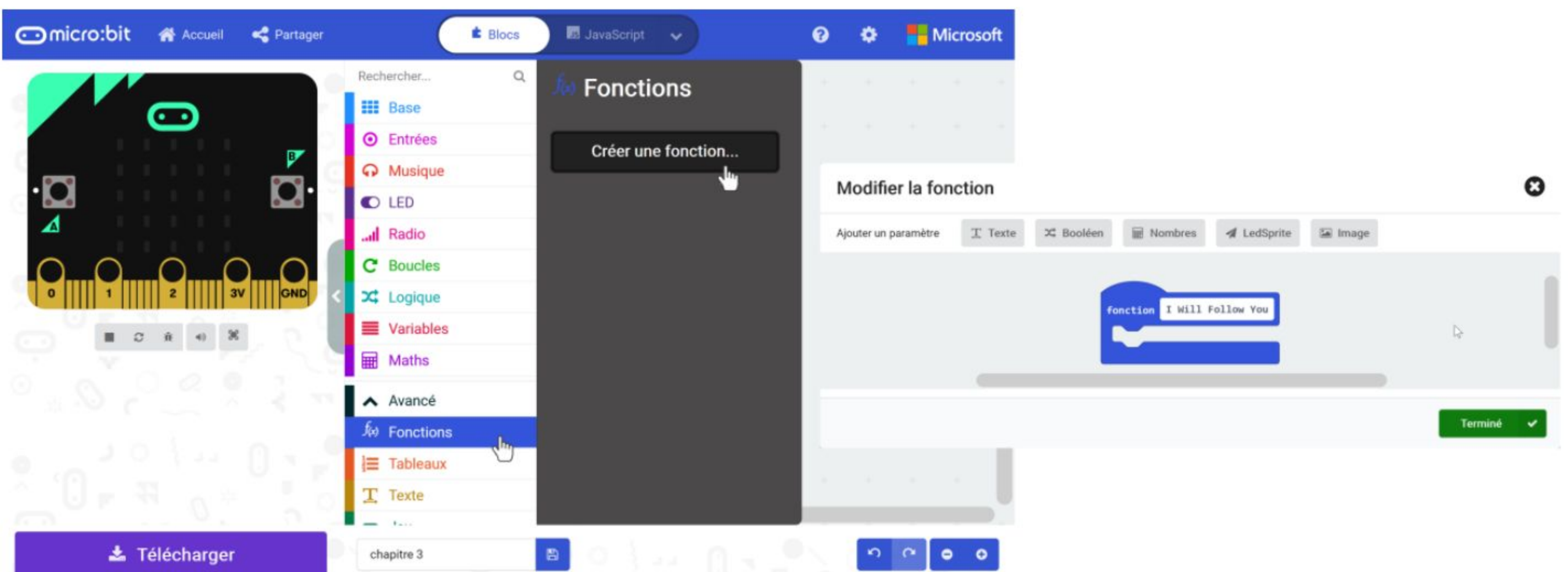


NOTE!

Nous pouvons créer des blocs de code qui réalisent des tâches spécifiques dans une **fonction**, comme par exemple ton code pour jouer la chanson "I Will Follow You".

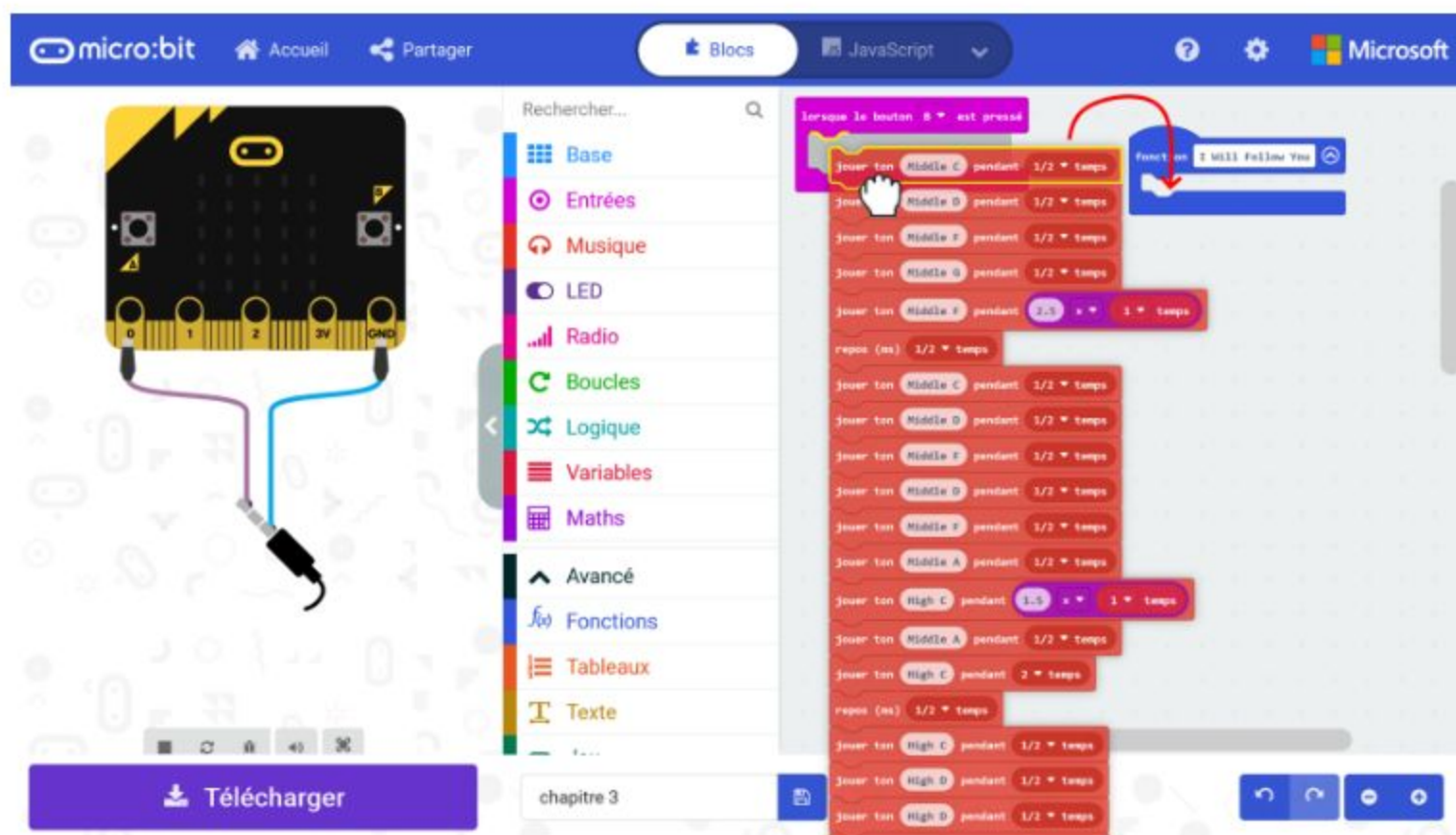
En programmation, une fonction fait référence à une méthode ou à un ensemble d'instructions. Une fois définie, la fonction peut être utilisée n'importe où dans ton programme, sans avoir à réécrire les mêmes blocs encore et encore.

Étape 11 Clique sur [**Avancé**] puis sur la catégorie [**Fonctions**]. Clique sur le bouton [**Créer une fonction...**] et renomme "faireQuelqueChose" en "**I Will Follow You**" dans la fenêtre pop-up. Ensuite, clique sur "**Terminé**".

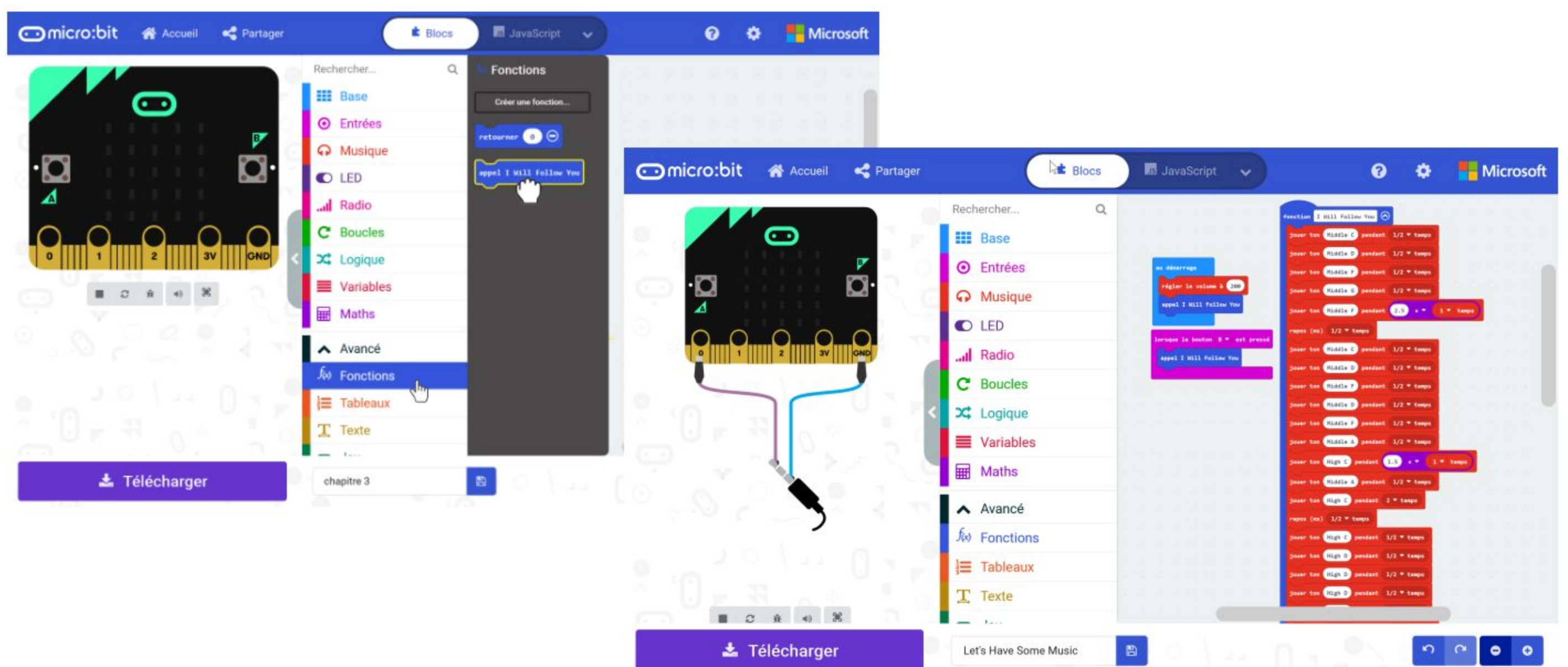


CHAPITRE 3 : Un peu de Musique~

Étape 12 Un bloc [fonction I Will Follow You] apparaît dans l'éditeur. Déplace l'ensemble des blocs contenus dans [lorsque le bouton B est pressé] en cliquant sur le premier d'entre-eux et imbrique-les dans le bloc [fonction I Will Follow You].



Étape 13 Clique sur la catégorie [Fonctions] et sélectionne le bloc [appel I Will Follow You]. Duplique-le. Imbrique les blocs [appel I Will Follow You] dans les blocs [au démarrage] et [lorsque le bouton B est pressé]. Voici une partie du code :



Étape 14 Flashe le code complet sur EDU:BIT. Apprécie la musique~

DÉCHIFFRE

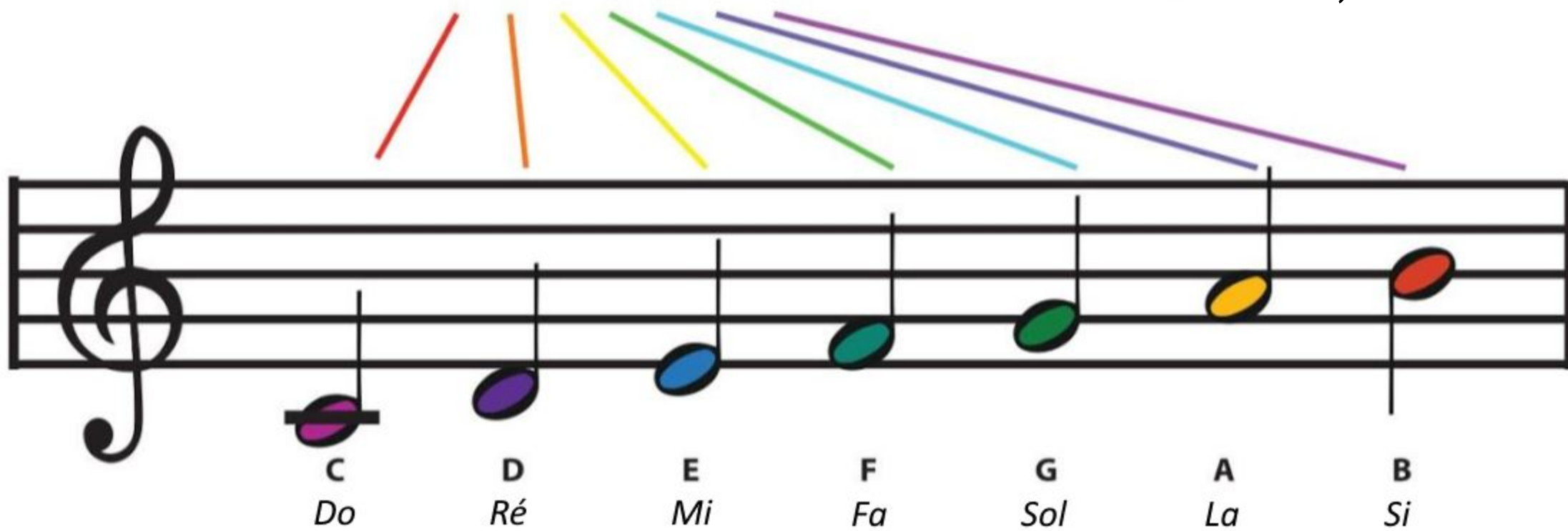
LE CODE






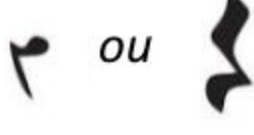







Tu peux programmer EDU:BIT pour jouer d'autres chansons si tu sais lire une partition. Voici un petit guide pour t'aider à "décoder" une partition de musique.



La position d'une note de musique sur la portée (ie : les 5 lignes horizontales) nous indique quelle tonalité jouer. La note la plus haute sur une partition est la note la plus aigue (fréquence la plus haute) et vice-versa.



Symbole	Nom	Durée	Silence
	Ronde	4 temps	
	Blanche	2 temps	
	Noire	1 temps	 ou 
	Croche	1/2 temps	
	Double croche	1/4 temps	

Différentes notations sont utilisées pour indiquer la durée d'une note (ie : combien de temps elle est jouée).

DÉCHIFFRE

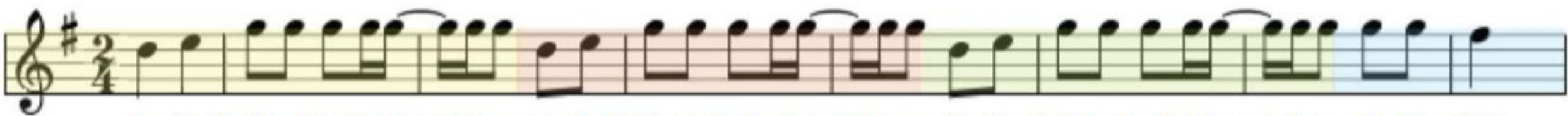
LE CODE

A l'aide des indices donnés, peux-tu "décoder" la mélodie suivante ?



Baby Shark

♩ = 115



Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark.

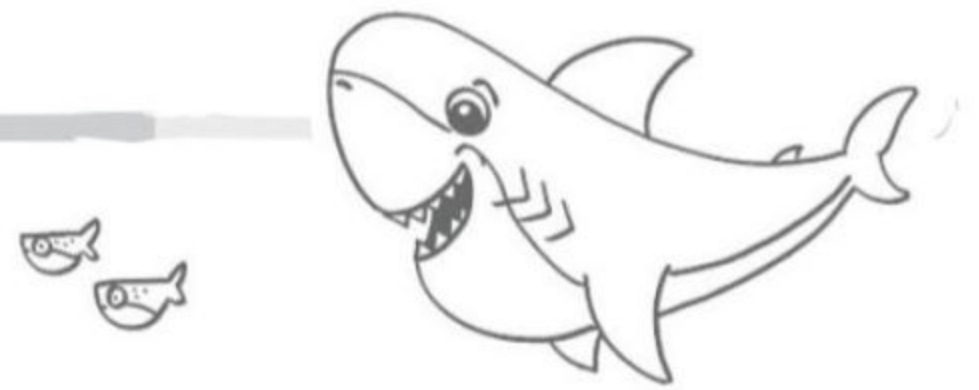
Line 1	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D	High E	High G		High G	High G	High G		High G
Beat	1		1/2	1/2	1/2		1/2	1/4	1/2

Line 2	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D		High G	High G	High G		High G	High G	High G
Beat	1/2	1/2	1/2	1/2	1/2	1/4	1/2		1/2

Répète la même chose pour la ligne 3

Line 4	Ba	-by	Shark
Note		High G	High F#
Beat	1/2	1/2	

Programme EDU:BIT afin qu'il joue l'air de Baby Shark lorsque que les boutons jaune (bouton A) et bleu (bouton B) sont pressés simultanément.



NOTE!

Utilise le bloc [régler le volume à _] pour régler le volume de la mélodie.



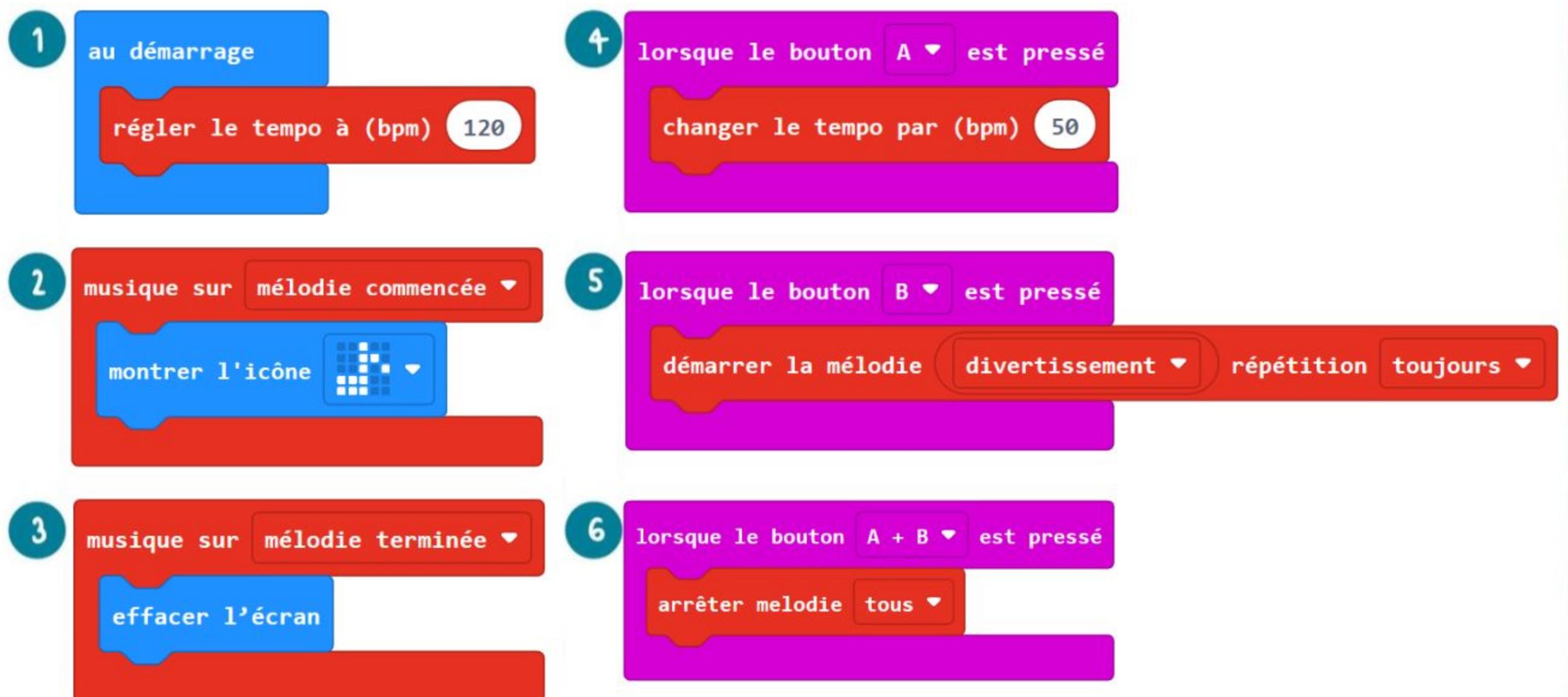
EXPLORE D'AUTRES BLOCS

#1 Tu peux définir le "tempo" (ie : le rythme de ta chanson) en utilisant le bloc [régler le tempo à (bpm) _]. Plus le bpm (battement par minute) est élevé, plus le rythme est rapide ou entraînant. Utilise le bloc [changer le tempo par (bpm) _] pour le modifier.

#2 Utilise le bloc [arrêter mélodie _] pour arrêter la mélodie en cours de lecture.

#3 Tu peux aussi utiliser [musique sur _] et sa liste de conditions possibles, telles que *mélodie commencée* et *mélodie terminée*, comme événements déclencheurs dans ton code.

Voici un exemple de code :

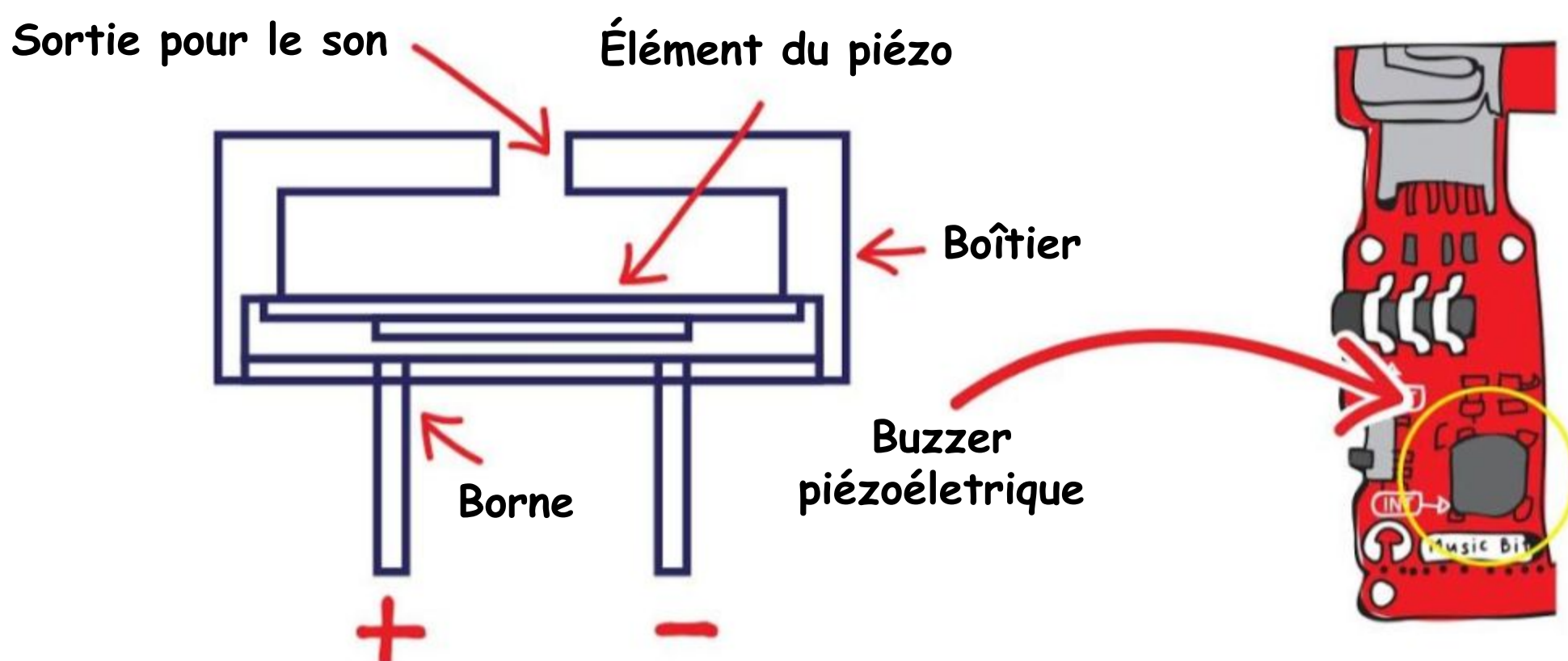


- 1 Dans ce programme, le tempo initial est défini à 120 bpm.
- 2 Lorsque *mélodie commencée*, une note de musique est affichée sur la matrice de LEDs.
- 3 Lorsque *mélodie terminée*, toutes les LEDs de la matrice sont éteintes.
- 4 Chaque fois que le bouton A est pressé, le tempo est incrémenté de 50 bpm.
- 5 Lorsque le bouton B est pressé, la mélodie "divertissement" est jouée.
- 6 Lorsque les boutons A+B sont pressés simultanément, toutes les mélodies sont arrêtées.

FAIT AMUSANT!

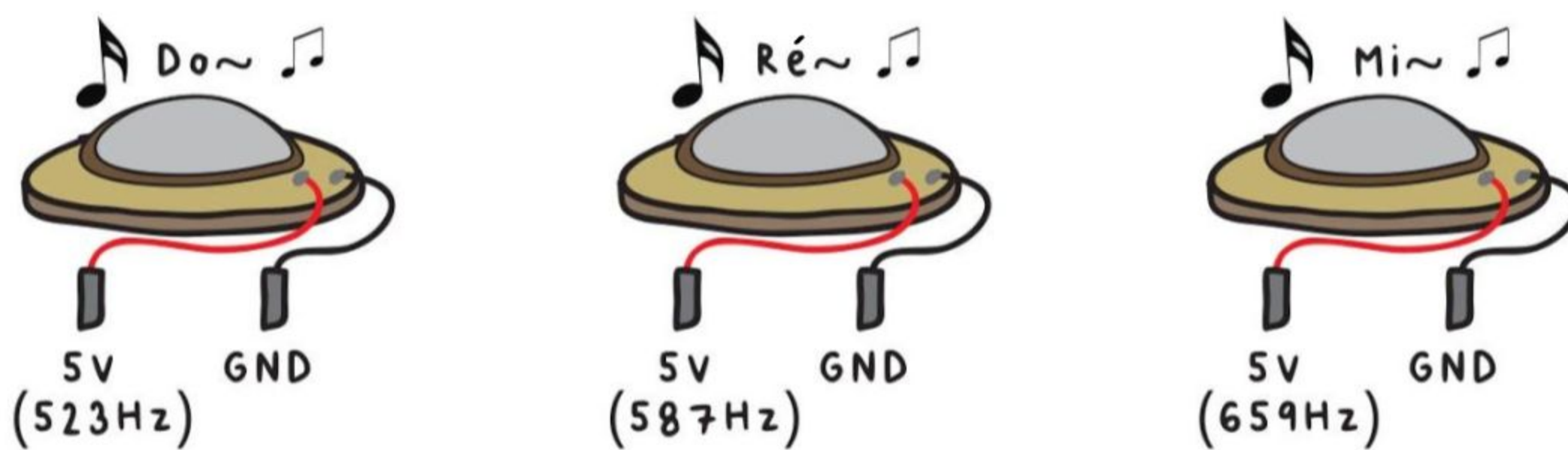


Un **buzzer piézoélectrique** est couramment utilisé pour produire des sons, par vibration d'un élément du piézo lorsqu'un signal électrique le traverse.



En changeant la fréquence du signal électrique, la vitesse de la vibration change. Par conséquent, le buzzer produit un son d'une tonalité différente.

Élément du piézoélectrique



L'oreille humaine peut entendre des sons dont la fréquence est comprise entre 20 Hz et 20 000 Hz. Un son dont la fréquence est inférieure à 20 Hz est appelé un infrason, tandis qu'un son dont la fréquence est supérieure à 20 000 Hz est appelé un ultrason.

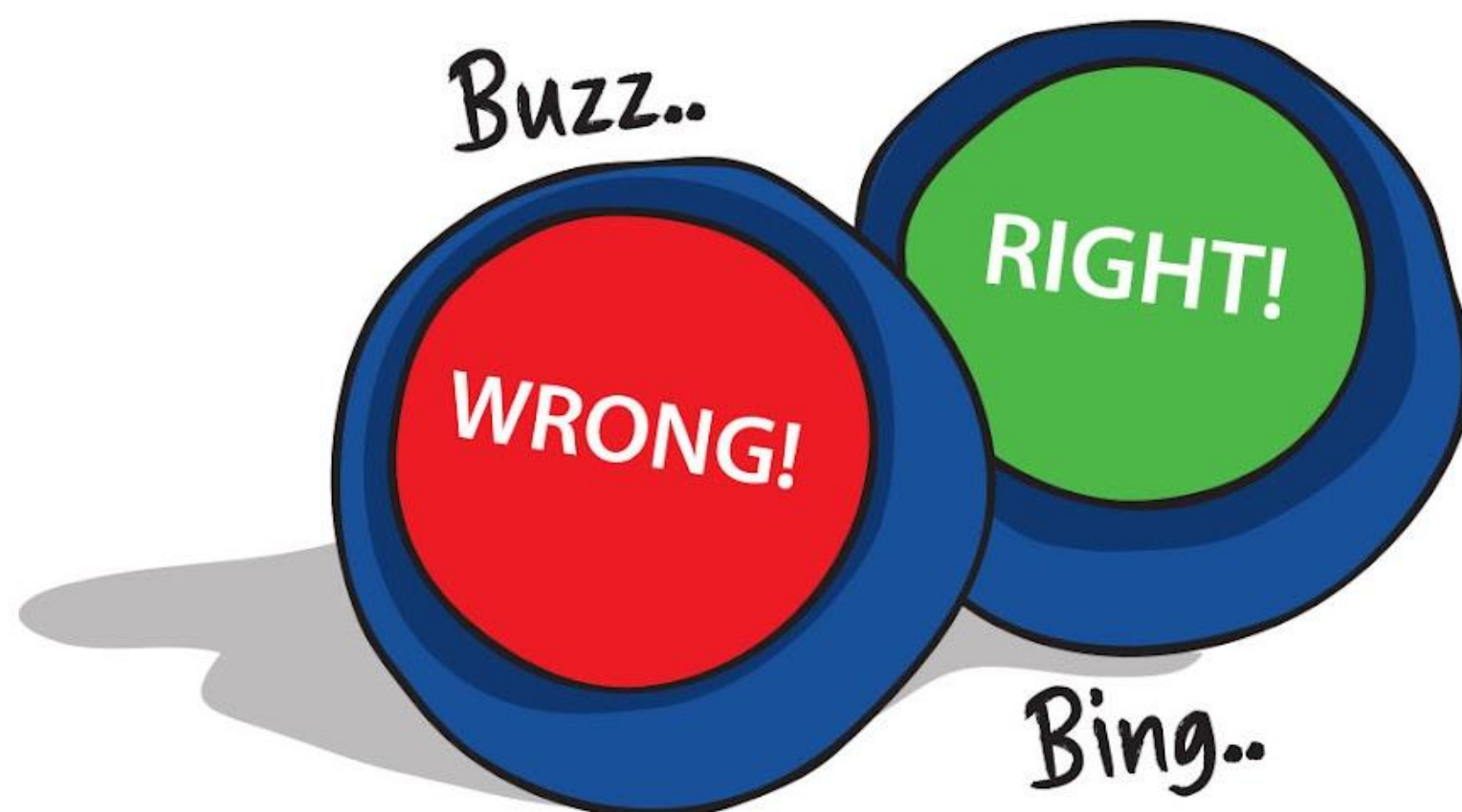


Apprends-en plus!



youtu.be/cxfPNc4Wefo

APPLICATION DÉFI



Programme EDU:BIT pour qu'il serve de buzzer pour un quizz, en indiquant si la réponse proposée est correcte ou erronée.

au démarrage	Afficher un visage <i>souriant</i> .
lorsque le bouton A est pressé (jaune).	Afficher l'icône '✓' et jouer la mélodie "mise sous tension" une fois.
lorsque le bouton B est pressé (bleu)	Afficher l'icône '✗' et jouer la mélodie "wawawaaa" une fois.
lorsque les boutons A+B sont pressés	Effacer l'écran d'affichage.

CHAPITRE 4

Dessinez, c'est Gagné !~

Traffic Light Bit



Une dinde !



Scanne-moi !




link.cytron.io/edubit-chapter-4

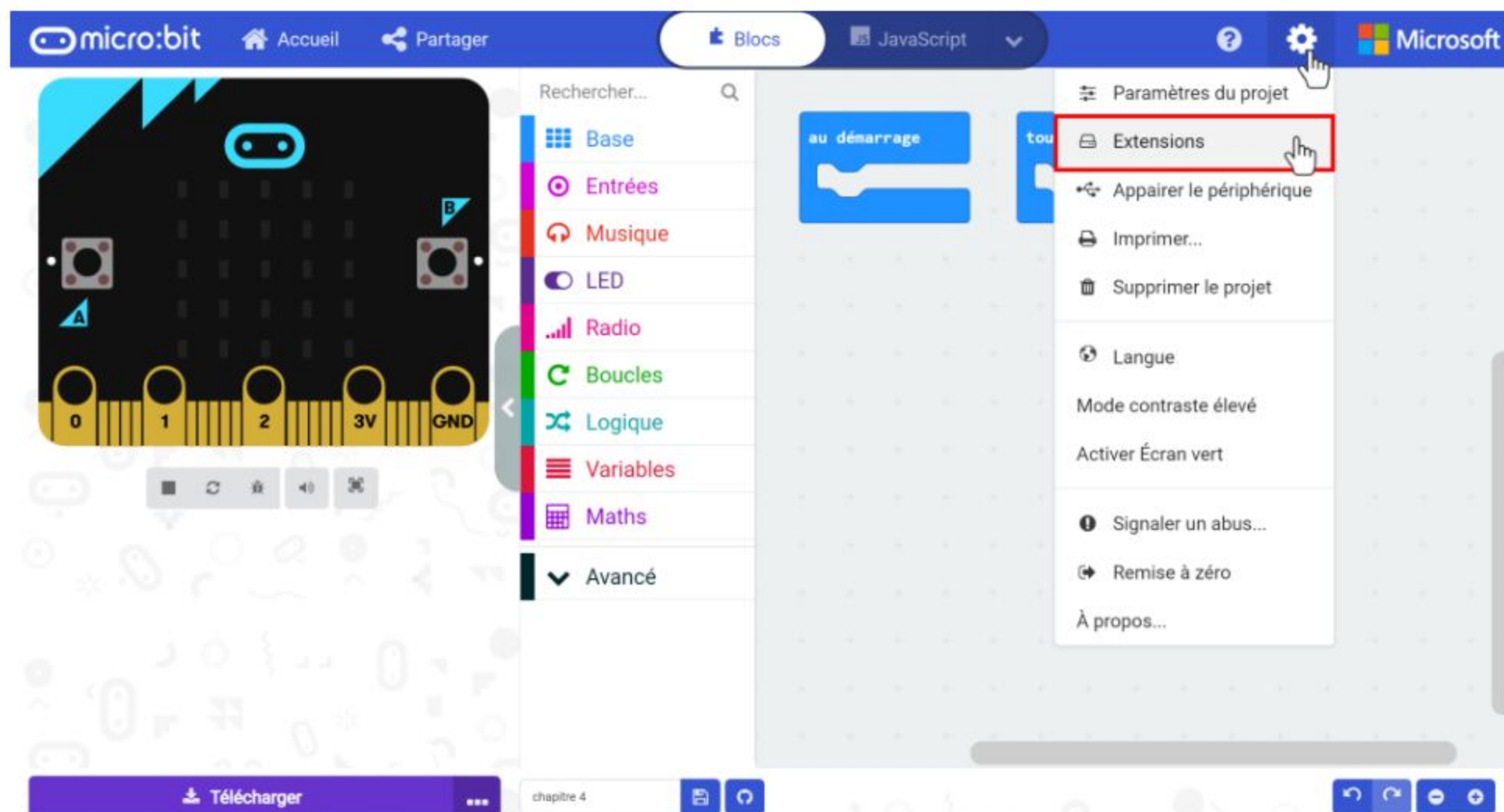


As-tu remarqué un ensemble de LEDs, rouge, jaune et verte sur EDU:BIT ? C'est le Traffic Light Bit. Pour le programmer, tu dois ajouter l'extension EDU:BIT à l'éditeur MakeCode. Les extensions sont des ensembles de blocs personnalisés que l'on peut ajouter à l'éditeur pour nous permettre de programmer facilement des modules complémentaires pour micro:bit, comme sur la carte EDU:BIT.

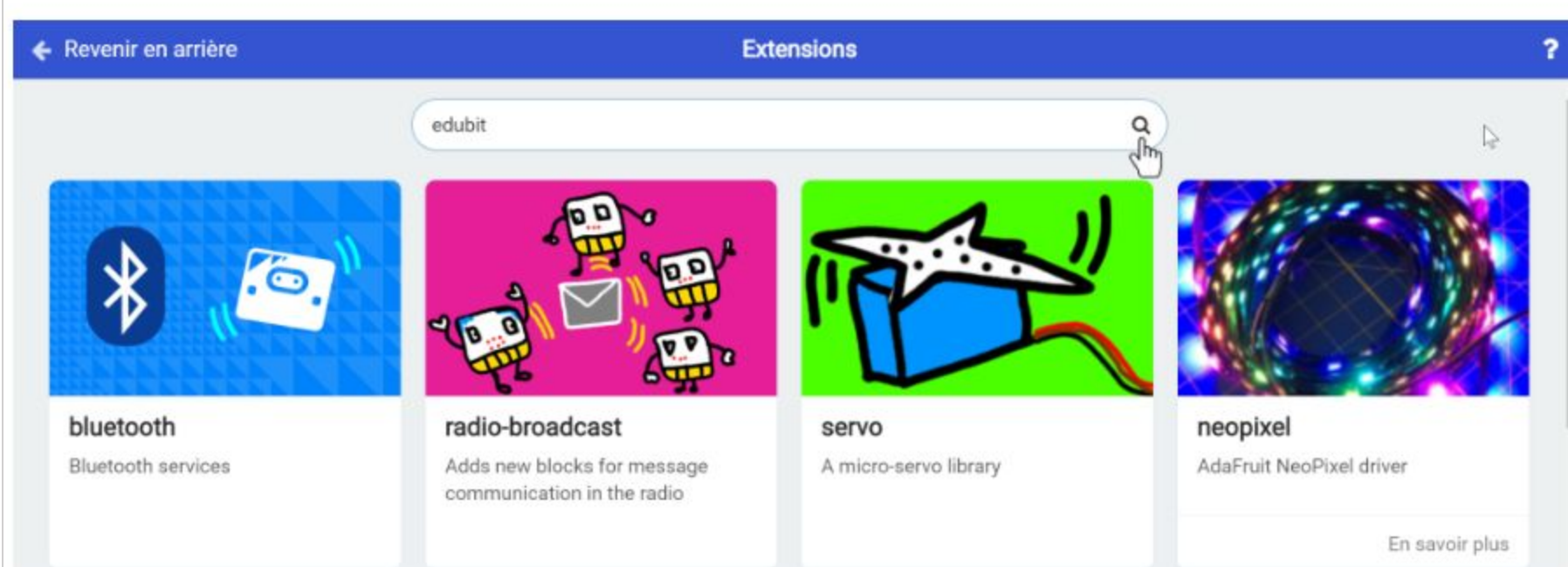


CODONS !

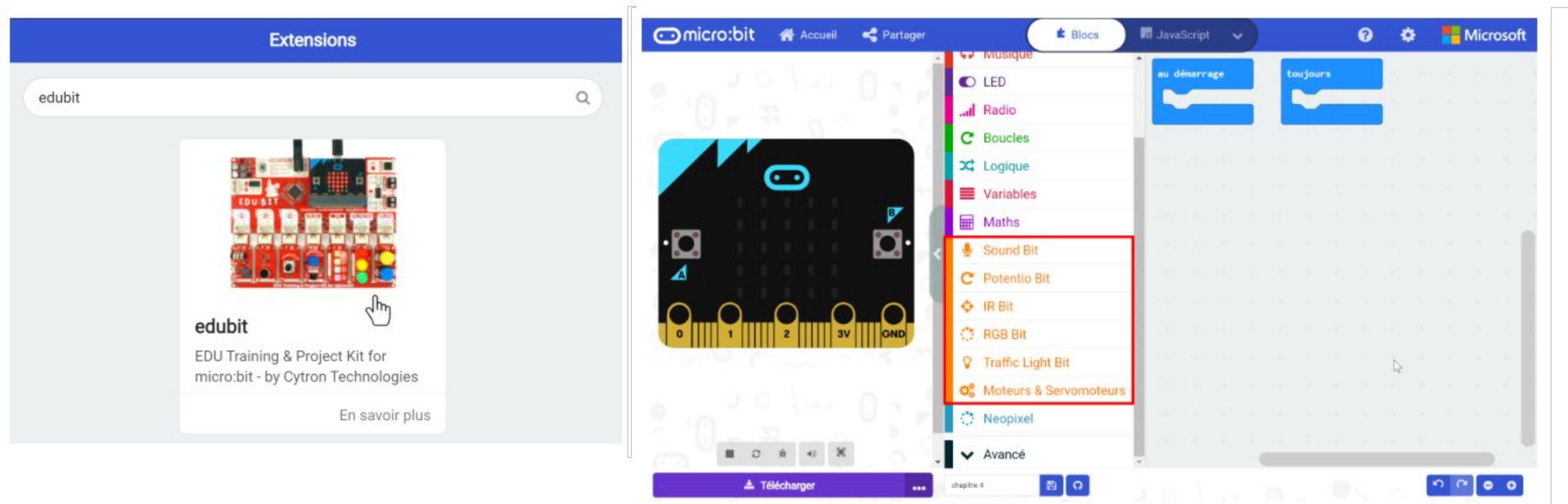
Étape 1 Crée un nouveau projet dans ton éditeur MakeCode. Clique sur la *roue dentée*  et sélectionne **“Extensions”**. *Tu as besoin d'une connexion Internet pour ajouter une extension.



Étape 2 Saisis **“edubit”** dans la zone de recherche et tape sur la touche Entrée.

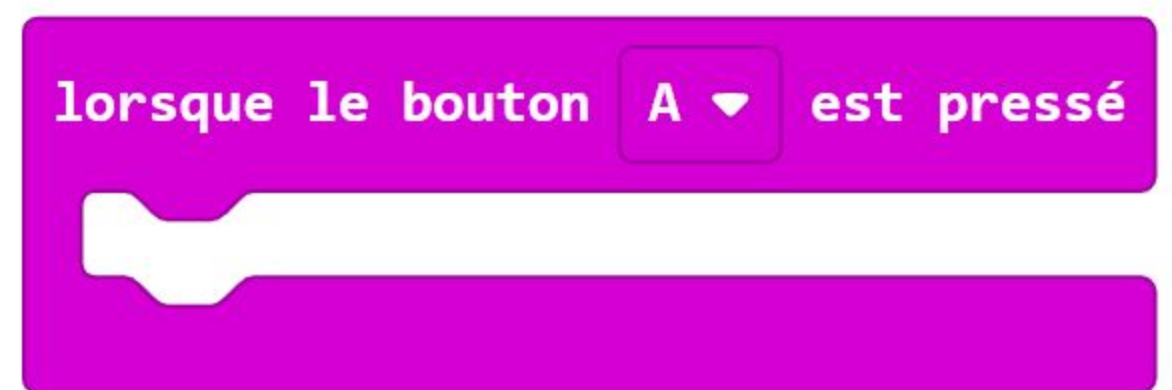
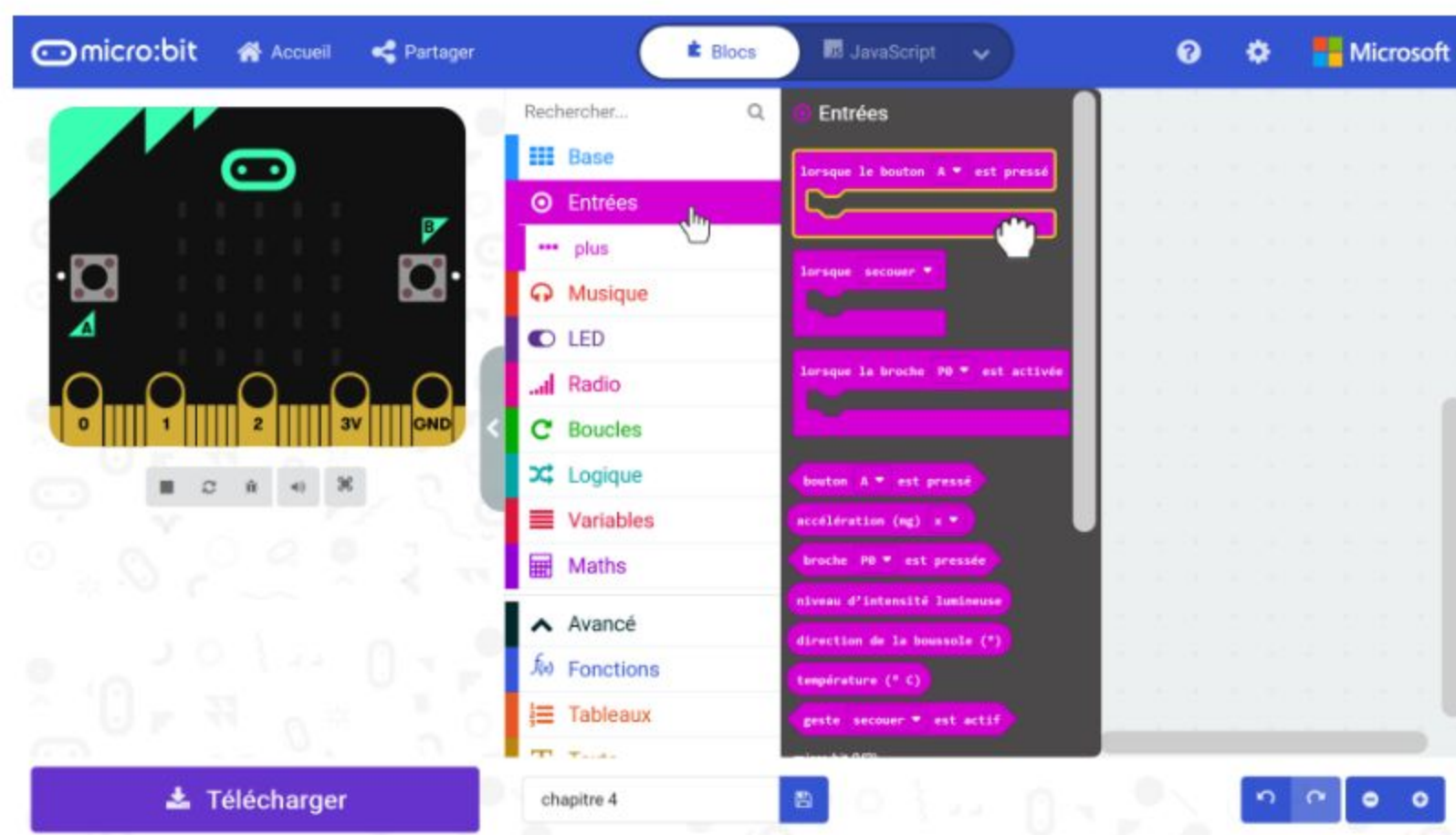


Étape 3 Clique sur l'extension **“edubit”**. Attends quelques instants pendant le chargement des nouvelles catégories dans l'éditeur MakeCode (voir ci-dessous).

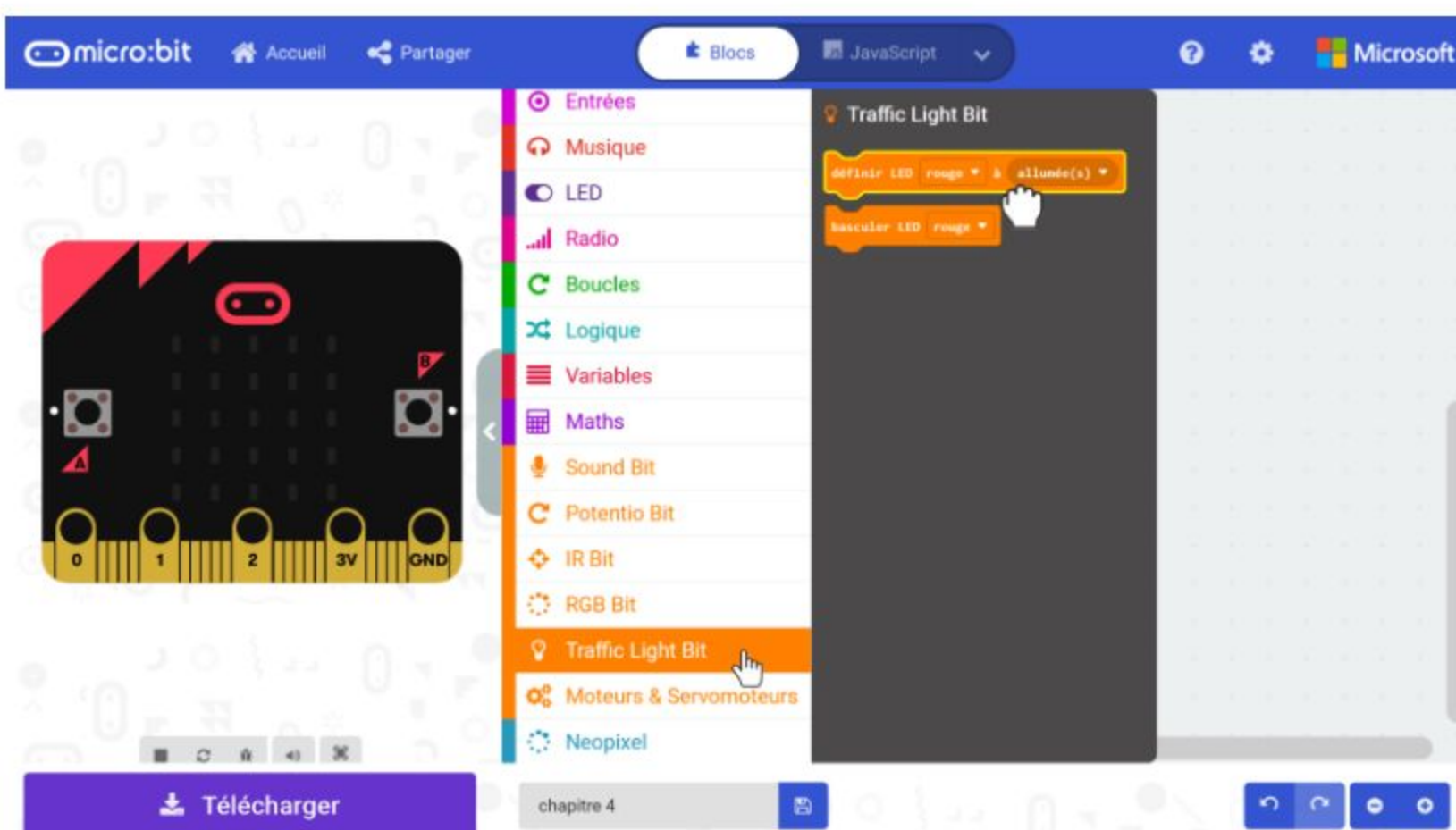


CHAPITRE 4 : Dessinez, c'est Gagné !~

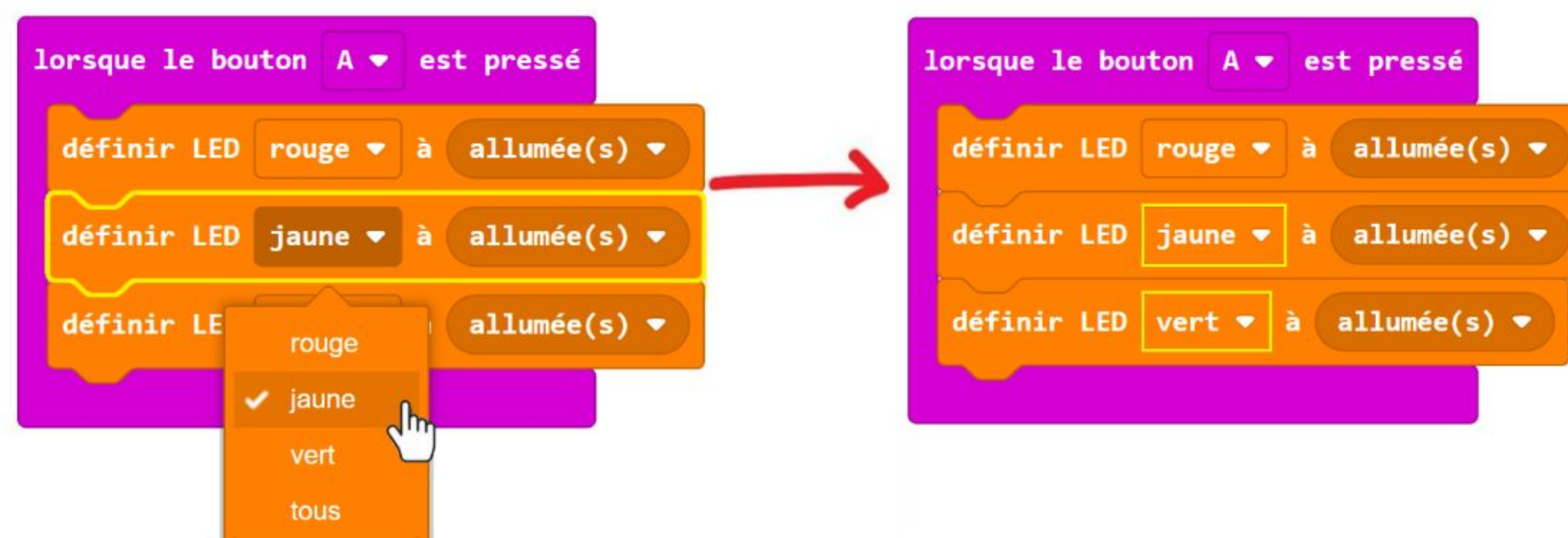
Étape 4 Cliquez sur la catégorie [**Entrées**] et sélectionnez le bloc [**lorsque le bouton _ est pressé**].



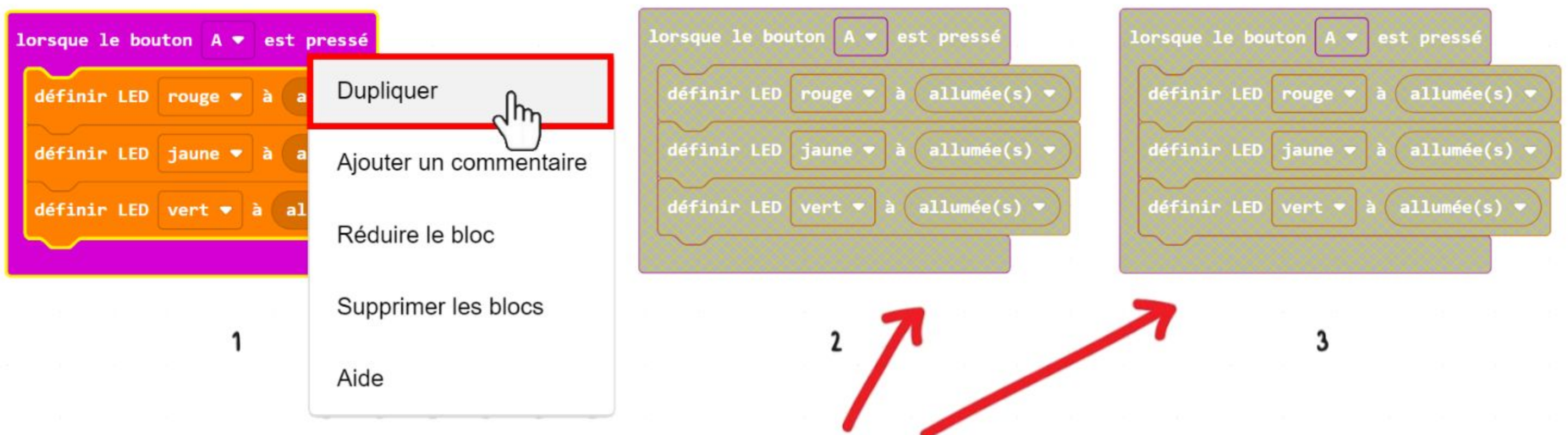
Étape 5 Cliquez sur la catégorie [**Traffic Light Bit**] et sélectionnez le bloc [**définir LED _ à _**]. Dans l'espace de travail, faites un clic droit sur le bloc [**définir LED _ à _**] et cliquez sur "Dupliquer". Répétez l'opération jusqu'à obtenir trois blocs [**définir LED _ à _**]. Imbriquez-les dans le bloc [**lorsque le bouton A est pressé**].



Étape 6 Cliquez sur le bouton de sélection de couleur et choisissez "jaune" pour le deuxième bloc, et "vert" pour le troisième bloc.



Étape 7 Fais un clic droit sur le bloc [**lorsque le bouton A est pressé**] et sélectionne "Dupliquer". Répète l'opération jusqu'à obtenir trois groupes de blocs identiques.



*Ces blocs sont désactivés et ne seront pas exécutés car il y a plusieurs blocs [**lorsque le bouton A est pressé**].

Étape 8 Dans les deuxième et troisième blocs [**lorsque le bouton A est pressé**], change "A" en "B" et "A+B" respectivement.

Étape 9 Change l'état des LEDs de allumée(s) à éteinte(s) comme ci-dessous.



Étape 10 Flashe le code sur EDU:BIT puis observe ce qu'il se passe lorsque tu appuies sur le bouton A, sur le bouton B, et enfin simultanément sur A+B.



RESTE DEHORS !

ATTENDS.

TU PEUX ENTRER.

Waouh... tu peux maintenant utiliser EDU:BIT comme ton indicateur personnel. Peux-tu imaginer d'autres utilisations ?



ÉVALUATION TRICOLEURE



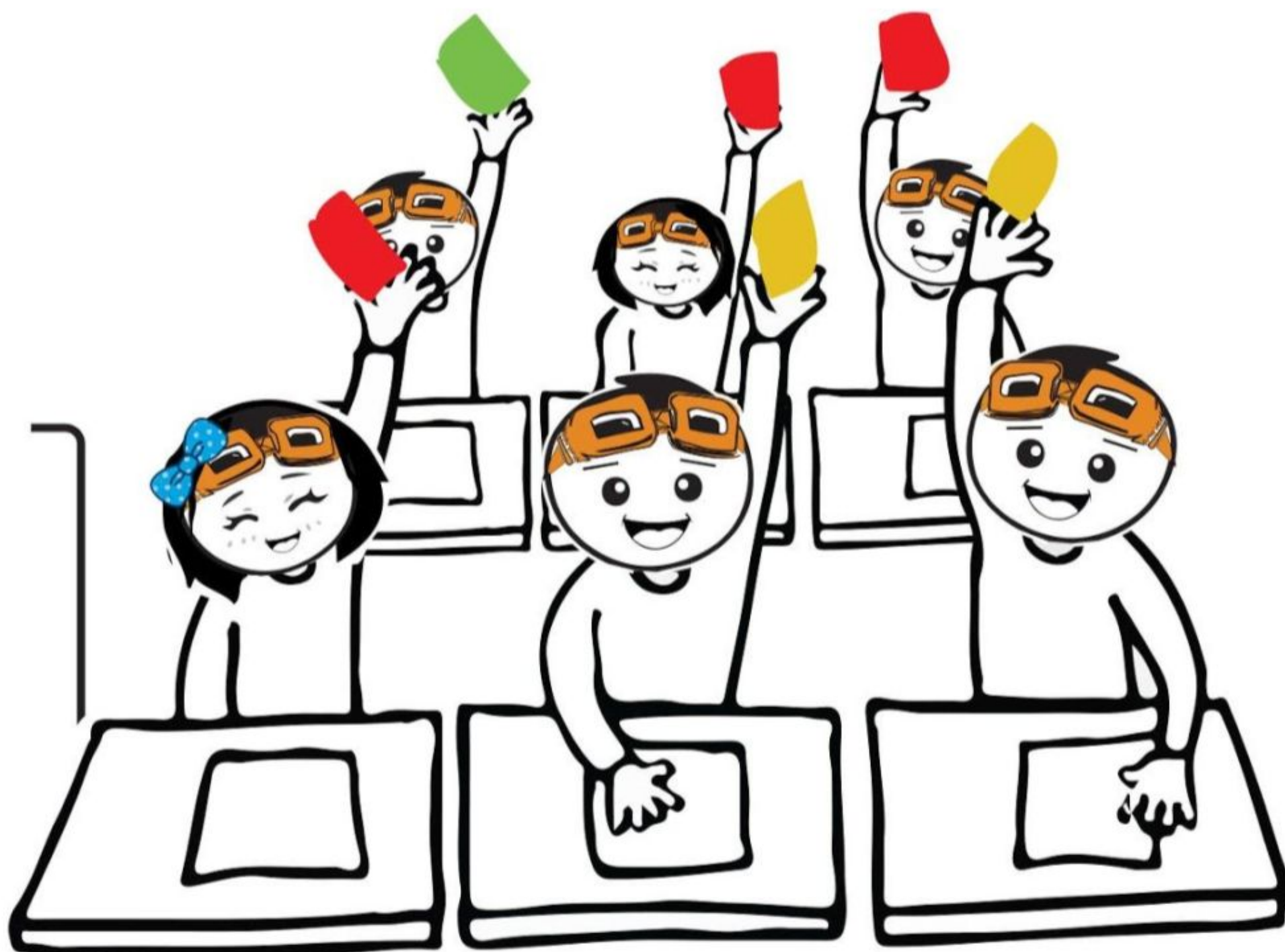
"Bloqué ! Aidez-moi."



"Je cherche encore."



"Compris !"



Une LED (ou DEL pour Diode ÉlectroLuminescente) est un exemple de périphérique de sortie. Elle ne possède que deux états : allumée et éteinte. On représente communément l'état "allumée" par le nombre 1 et "éteinte" par le nombre 0.



Tu peux aussi programmer EDUBIT pour qu'il fonctionne comme un chronomètre. Voici un exemple de code.



- Le chronomètre est déclenché lorsque tu secoues EDU:BIT.
- Joue un son pour signaler que le chronomètre a démarré.
- La LED verte s'allume.
- La LED jaune s'allume.
- La LED rouge s'allume.
- Joue la mélodie Wawawawaa pour signaler que le temps est écoulé.
- Fait clignoter la LED rouge 10 fois.

```

lorsque secouer
  jouer ton Middle C pendant 1 temps
  définir LED rouge à éteinte(s)
  définir LED jaune à éteinte(s)
  définir LED vert à allumée(s)
  pause (ms) 2000
  définir LED rouge à éteinte(s)
  définir LED jaune à allumée(s)
  définir LED vert à éteinte(s)
  pause (ms) 2000
  définir LED rouge à allumée(s)
  définir LED jaune à éteinte(s)
  définir LED vert à éteinte(s)
  pause (ms) 2000
  démarrer la mélodie wawawawaa répétition une fois
  répéter 10 fois
    faire
      basculer LED rouge
      pause (ms) 500
  
```

Dans cet exemple de code, chaque LED s'allume pendant 2 000 ms (2 seconds).
Si tu avais voulu que chaque LED s'allume pendant 1 minute, quelle valeur aurais-tu dû saisir ici ?
Voici une indication :
1 minute = 60 secondes

```

basculer LED rouge
  
```

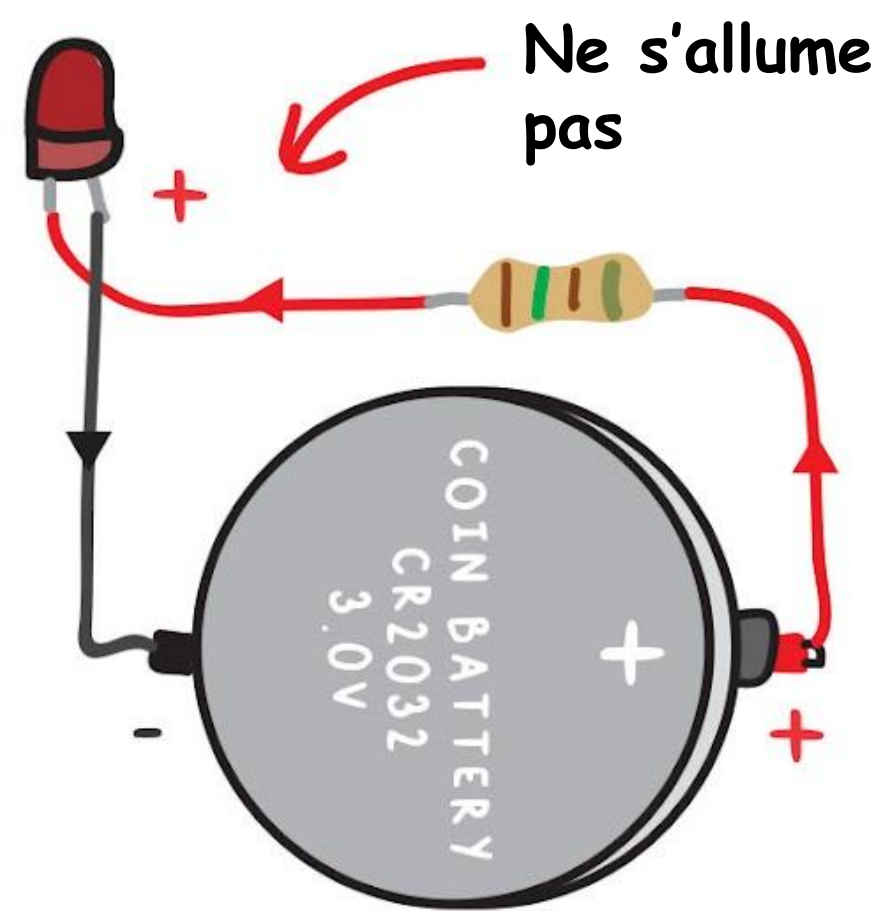
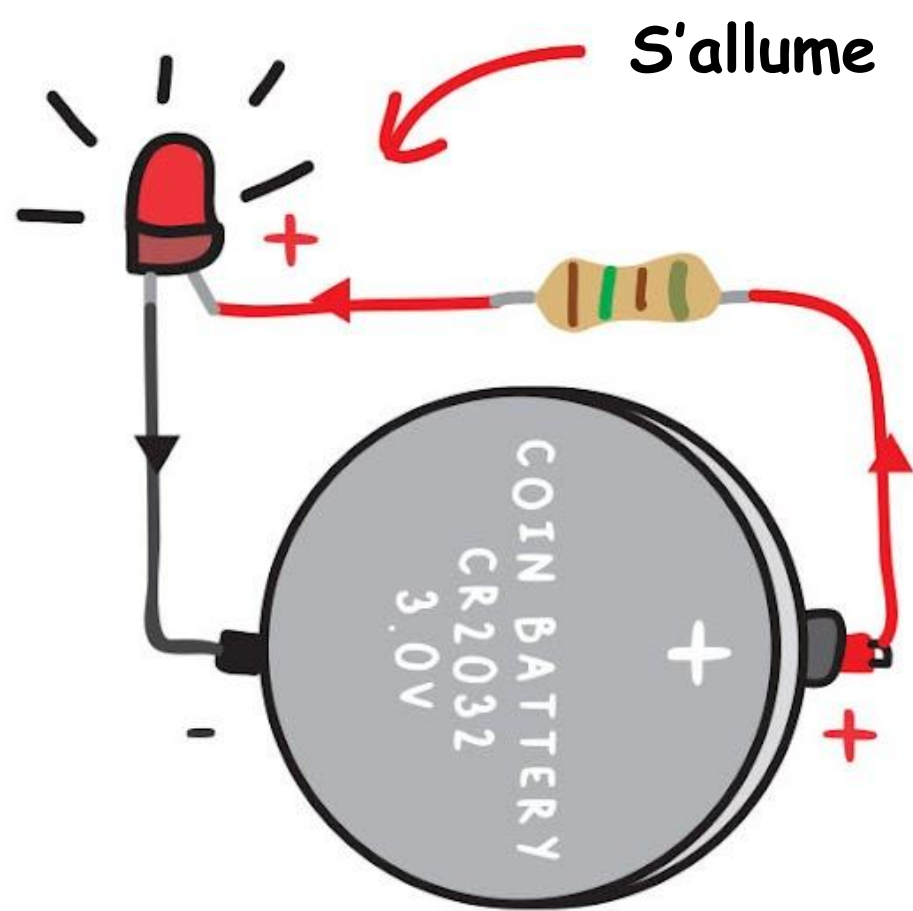


"Basculer" signifie passer d'un état à l'autre. Si l'état est "allumée", alors il deviendra "éteinte", et inversement. Ainsi, si on répète l'action "basculer" sur une même LED, elle semblera clignoter.

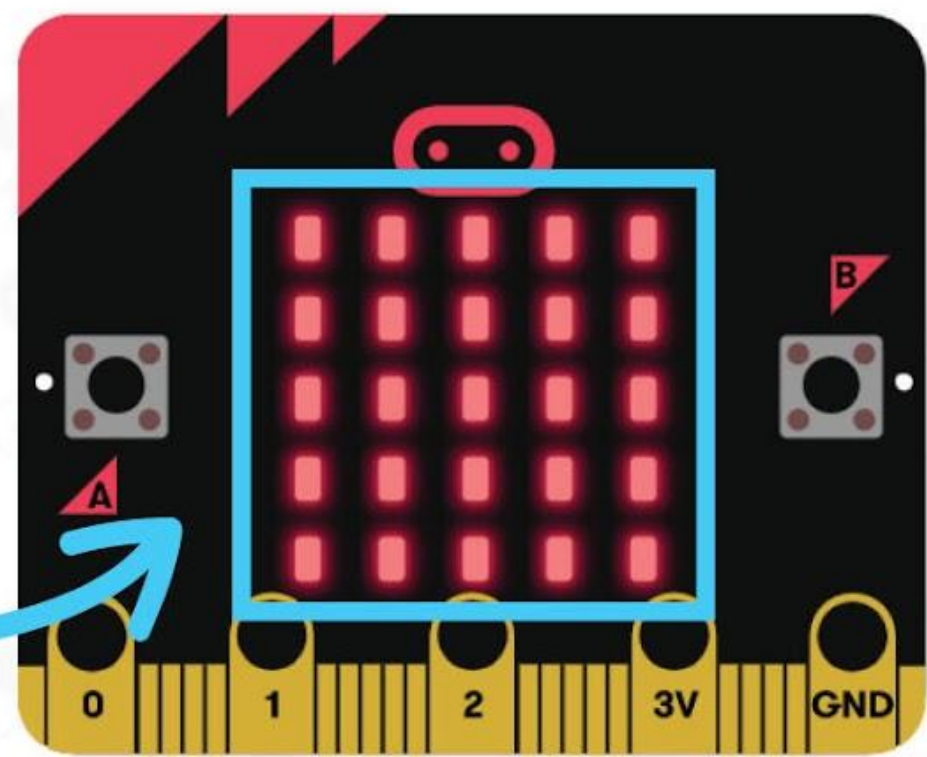
FAIT AMUSANT!



Une **diode électroluminescente (LED)** est un composant semi-conducteur qui produit de la lumière à partir d'électricité. Elle possède deux bornes : une borne positive et une borne négative. Lorsqu'une LED est connectée en respectant la bonne polarité et qu'elle est traversée par un courant, elle émet de la lumière.



Les LEDs utilisées sur le microbit sont des composants montés en surface (SMT : Surface Mount Technology) qui peuvent être très petits.



En plus de celles du microbit, il y a 41 autres LEDs montées en surface sur EDU:BIT. Peux-tu toutes les retrouver ?



APPLICATION DÉFI

Programme EDU:BIT pour qu'il serve de minuteur et de compteur de scores pour des jeux tels que "Dessinez, c'est gagné !" ou des "charades mimées".	
au démarrage	Initialiser la variable <i>équipe A</i> à 0. Initialiser la variable <i>équipe B</i> à 0.
lorsque le bouton A est pressé (jaune)	Incrémenter <i>équipe A</i> de 1. Afficher le score actuel de l'équipe A.
lorsque le bouton B est pressé (bleu)	Incrémenter <i>équipe B</i> de 1. Afficher le score actuel de l'équipe B.
lorsque les boutons A+B sont pressés	Afficher les scores d'équipe A et d'équipe B.
lorsque secouer	Lancer le minuteur pendant 1 minute en allumant la LED verte (pendant 30 secondes), puis la LED jaune (pendant 20 secondes) et enfin la LED rouge (pendant 10 secondes). Jouer la mélodie "wawawaaa" lorsque le temps est écoulé. La LED rouge clignote 10 fois.

*Voici une indication pour toi. Tu dois créer deux variables et les nommer *équipe A* et *équipe B* respectivement.*



Jouons

Dessinez, c'est Gagné !~



COMMENT JOUER :

- Diviser la classe en deux équipes : équipe A et équipe B.
- Un membre de l'équipe A commence par tirer une carte au hasard. Après avoir lu le mot ou la phrase sur la carte, secouer EDU:BIT pour lancer le minuteur.
- Il / Elle peut ensuite commencer à dessiner des images sur le tableau pour faire deviner le mot aux autres membres de l'équipe. Aucun mot ni geste ne sont autorisés.
- L'équipe gagne un point (appuyer sur le bouton A ou le bouton jaune) si l'un des membres de l'équipe trouve le mot ou la phrase avant la fin du temps imparti.
- En cas d'échec, l'équipe B peut tenter de récupérer le point en proposant une réponse.
- On poursuit en échangeant les rôles, et ainsi de suite jusqu'à la fin de la partie.
- L'équipe ayant remporté le plus de points gagne !

NOTE!

Scanne ici !

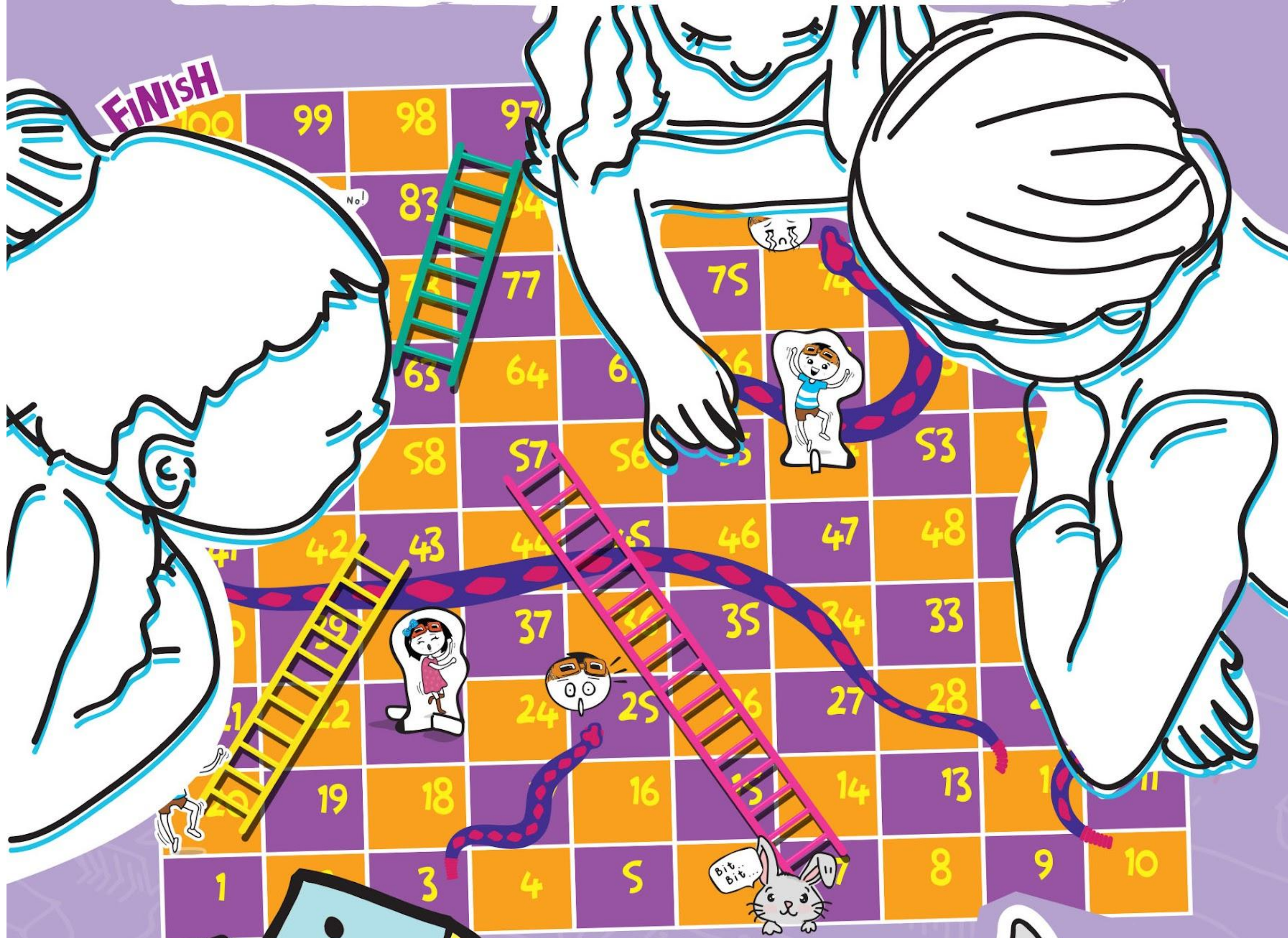


Scanne le QR code pour télécharger des cartes de mots à deviner imprimables. Si dessiner n'est pas ton truc, tu peux essayer les charades mimées. Les règles sont les mêmes mais au lieu de dessiner, tu dois faire des gestes pour faire deviner le mot aux membres de ton équipe. Amuse-toi !

CHAPITRE 5

Dé Numérique... Roule~

IR Bit



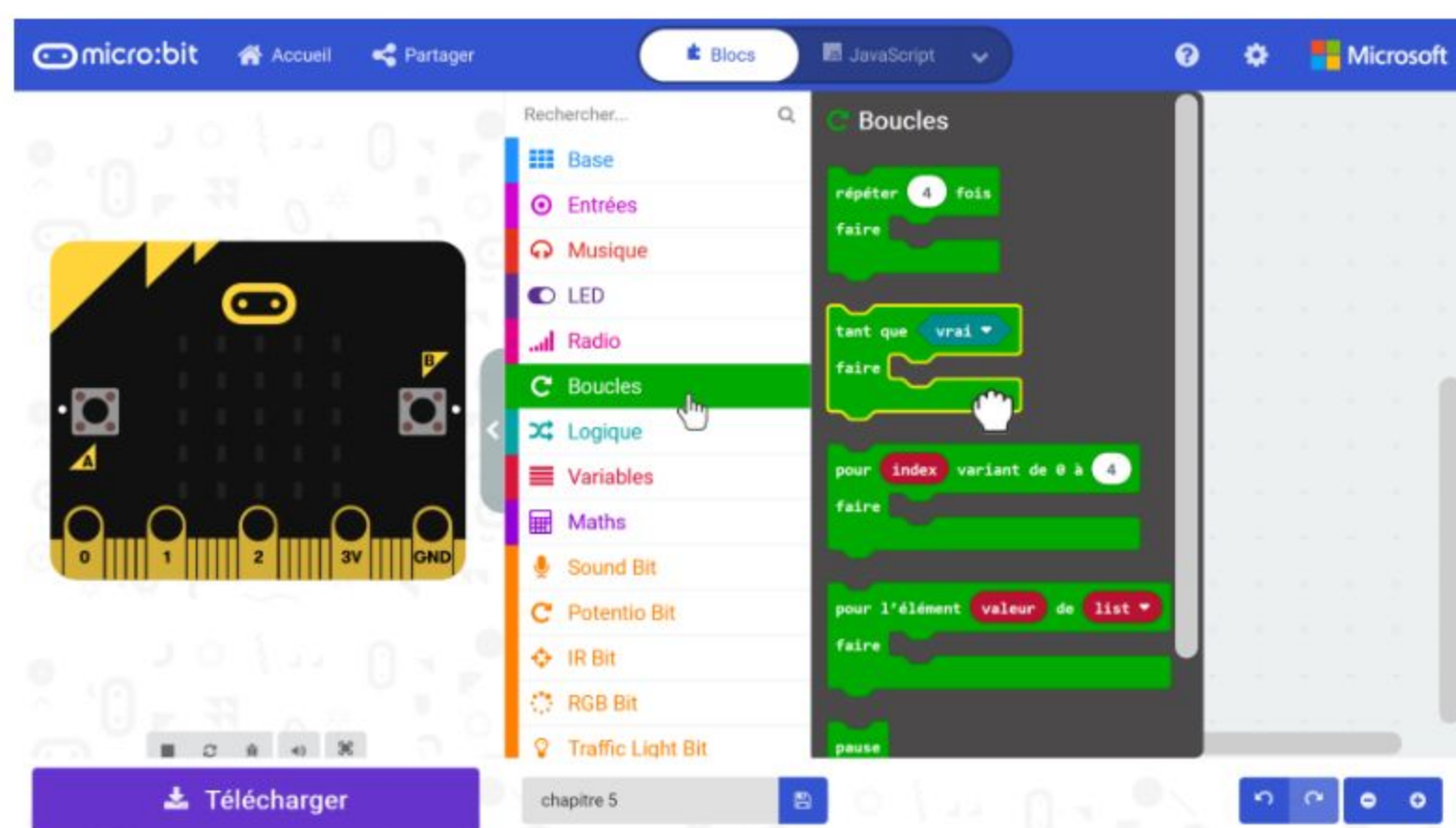
Scanne-moi !



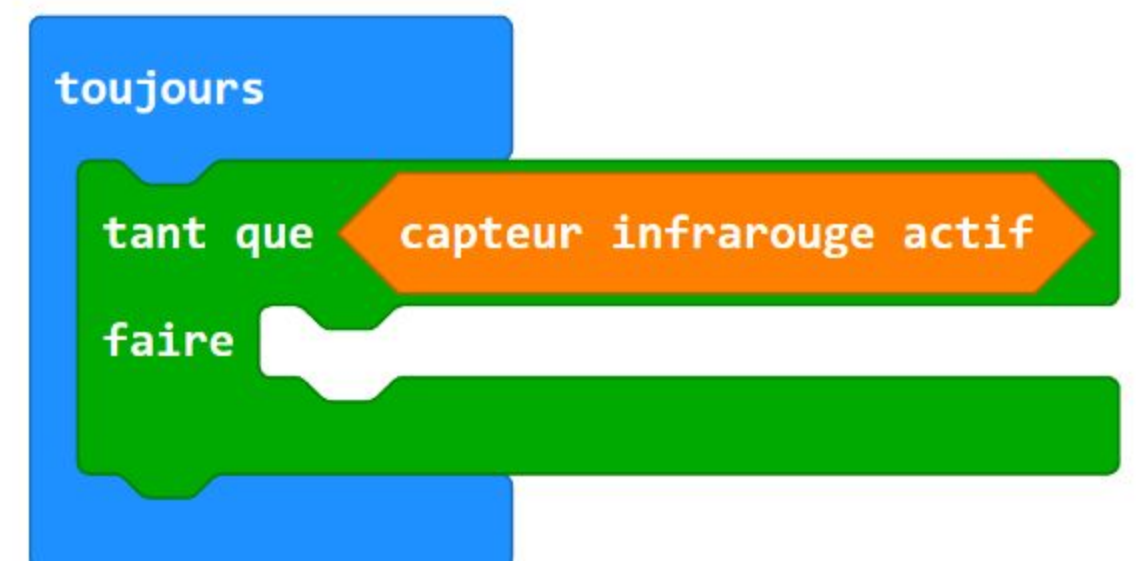
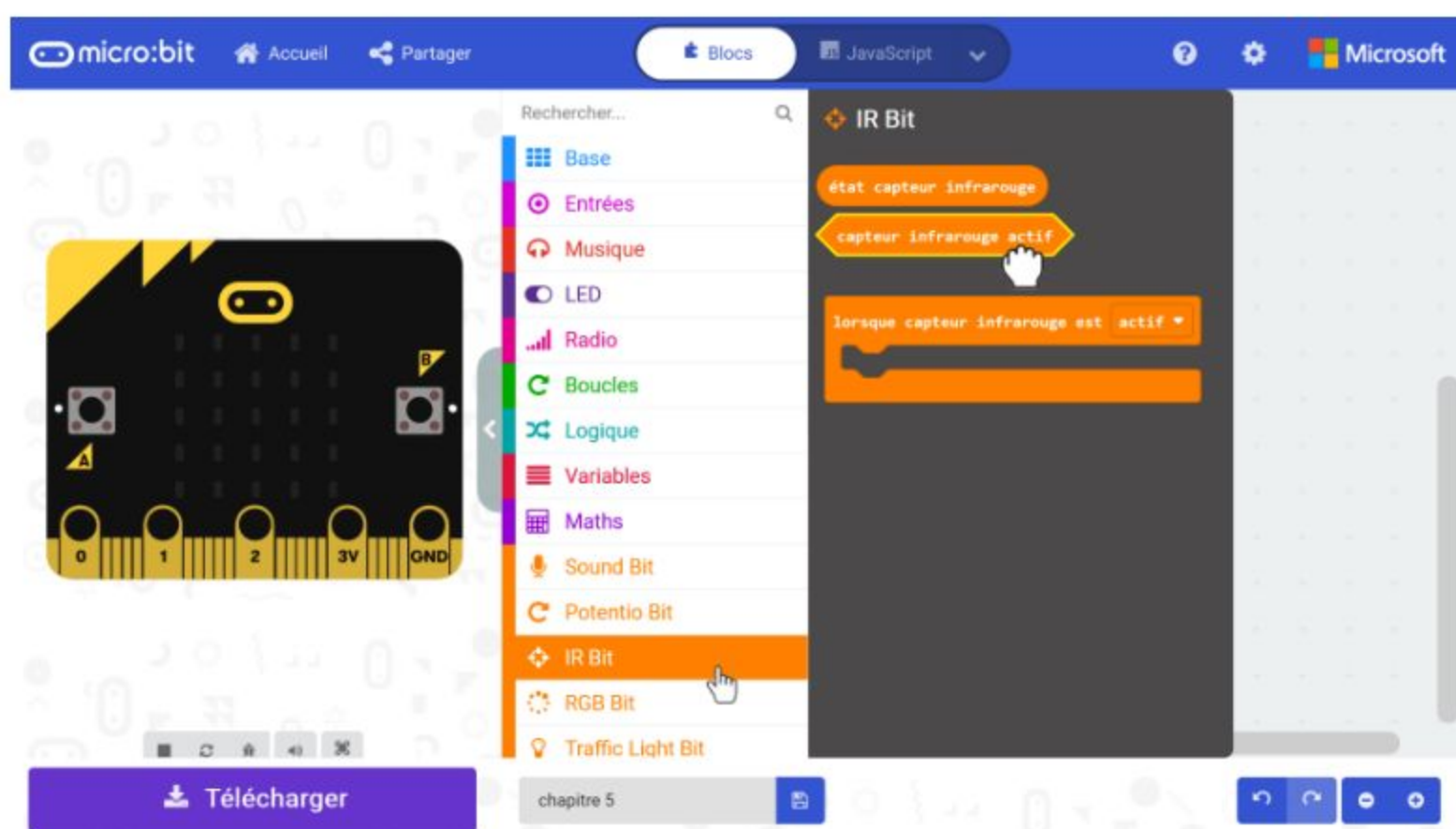
link.cytron.io/edubit-chapter-5

Codons !

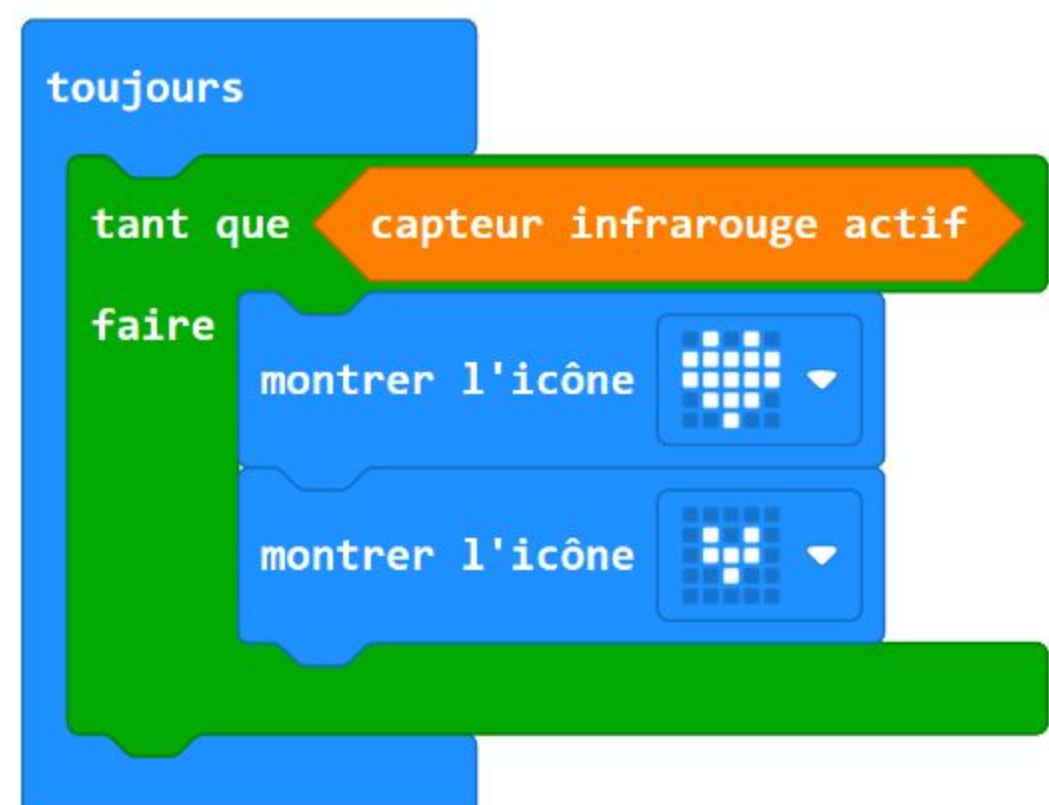
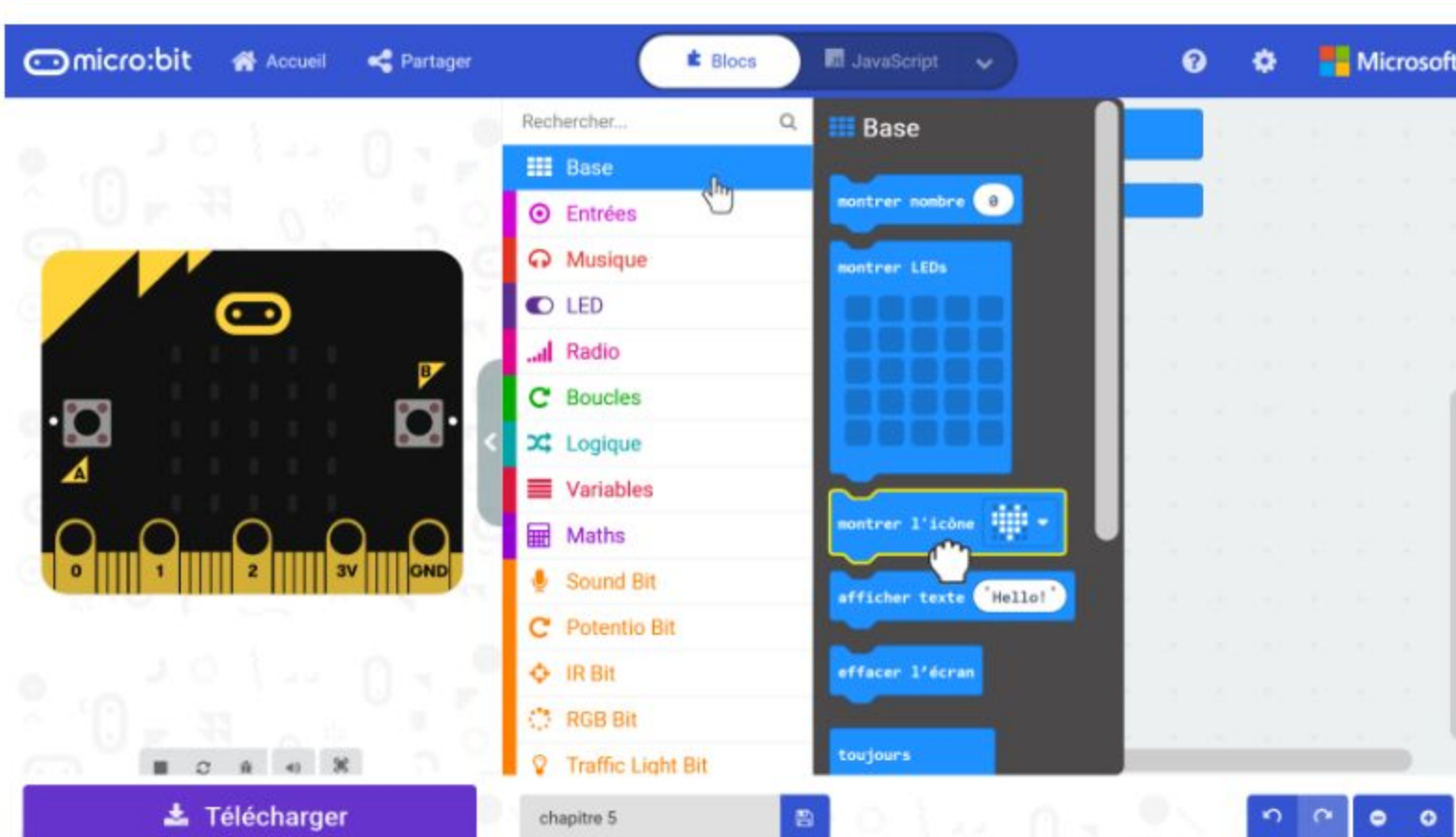
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT (voir page 40). Clique sur la catégorie [**Boucles**] et sélectionne le bloc [**tant que _ faire**]. Imbrique-le dans le bloc [**toujours**].



Étape 2 Clique sur le catégorie [**IR Bit**] et sélectionne le bloc [**capteur infrarouge actif**]. Insère-le dans la condition du bloc [**tant que _ faire**].

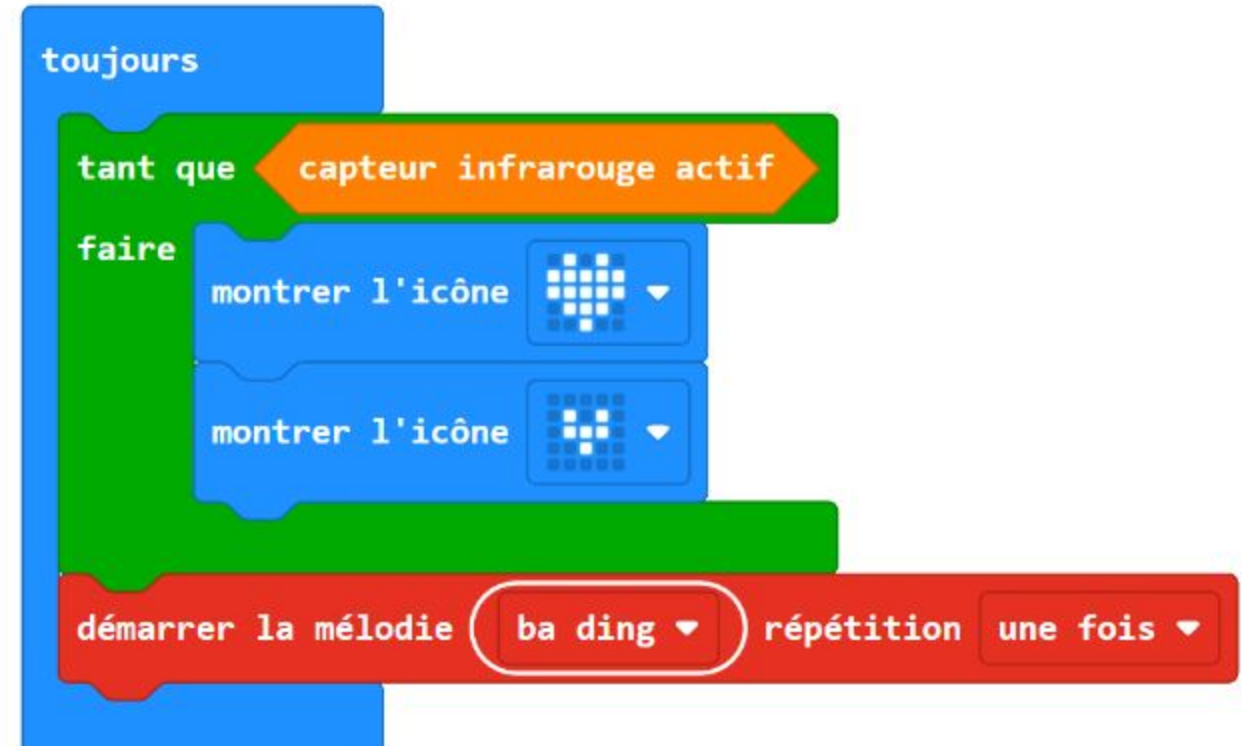
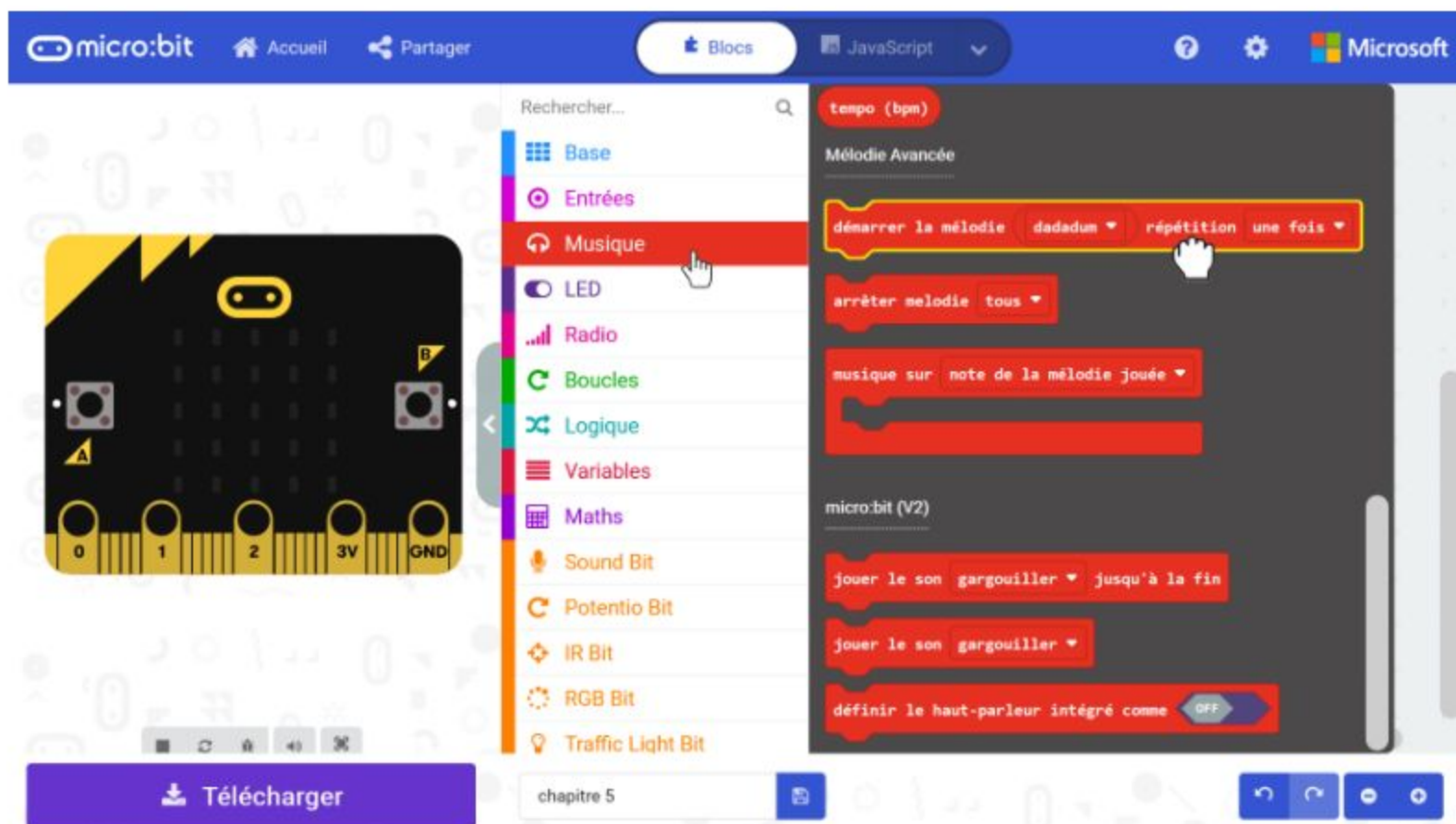


Étape 3 Clique sur la catégorie [**Base**] et ajoute deux blocs [**afficher l'icône**]. Change l'une des deux icônes en "petit cœur". Imbrique-les dans le bloc [**tant que _ faire**].

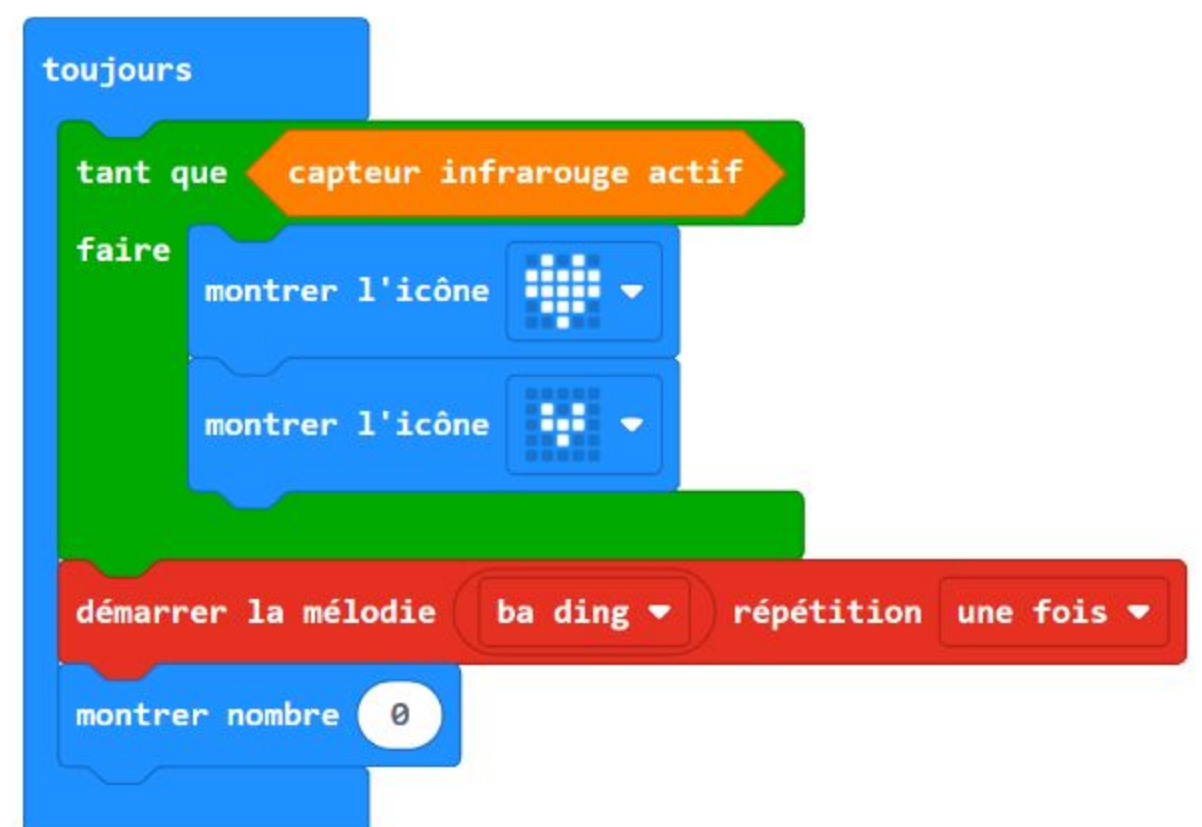
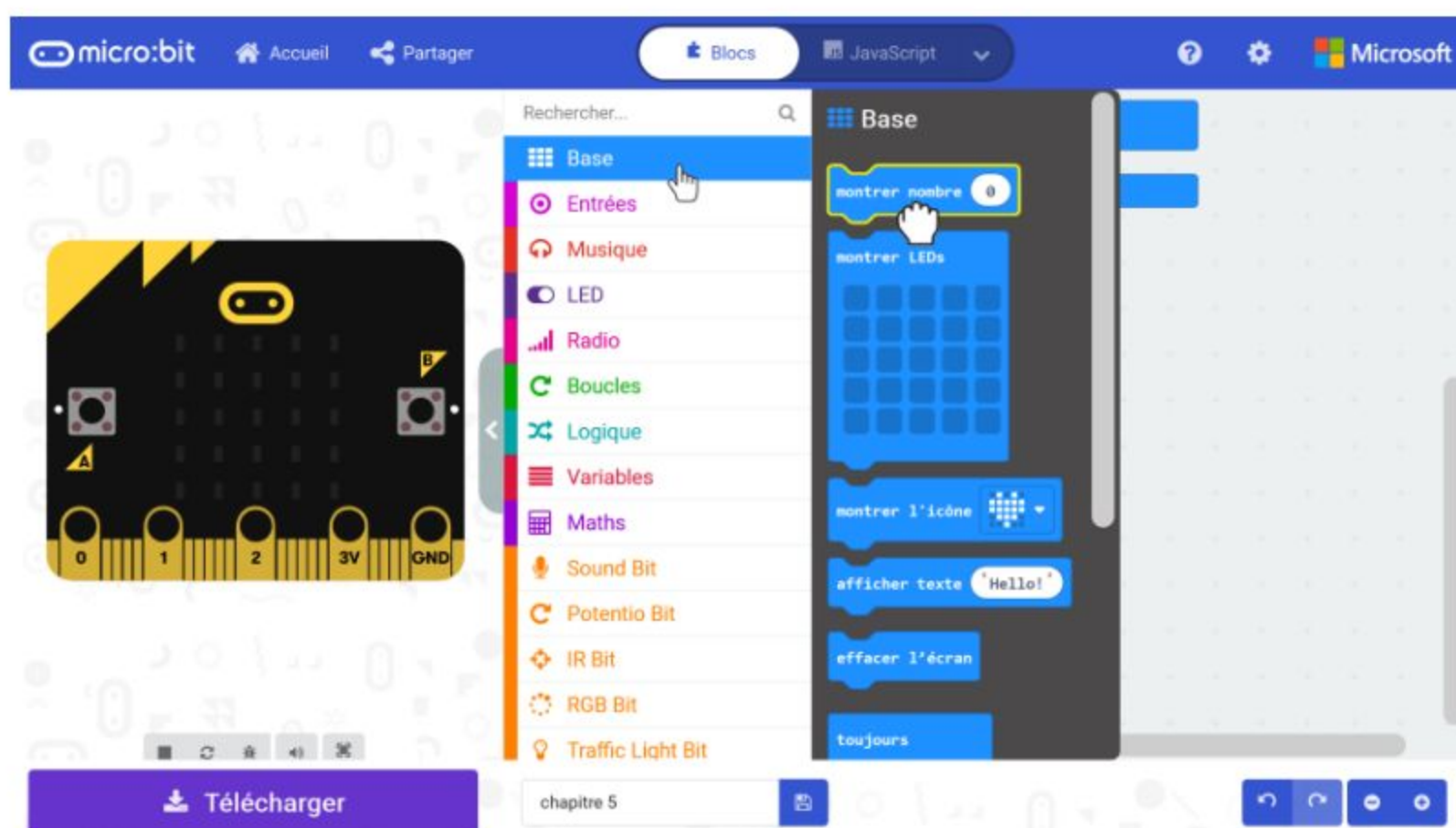




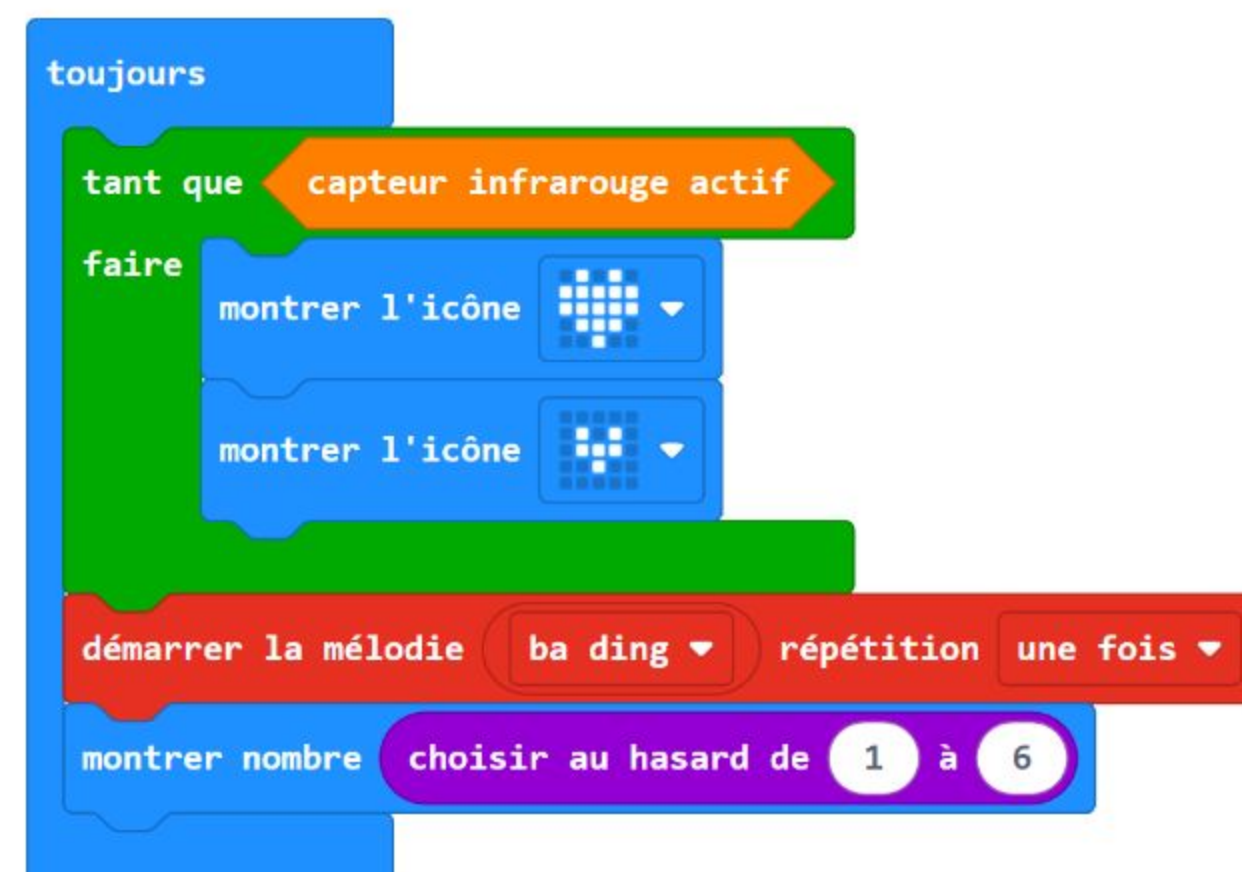
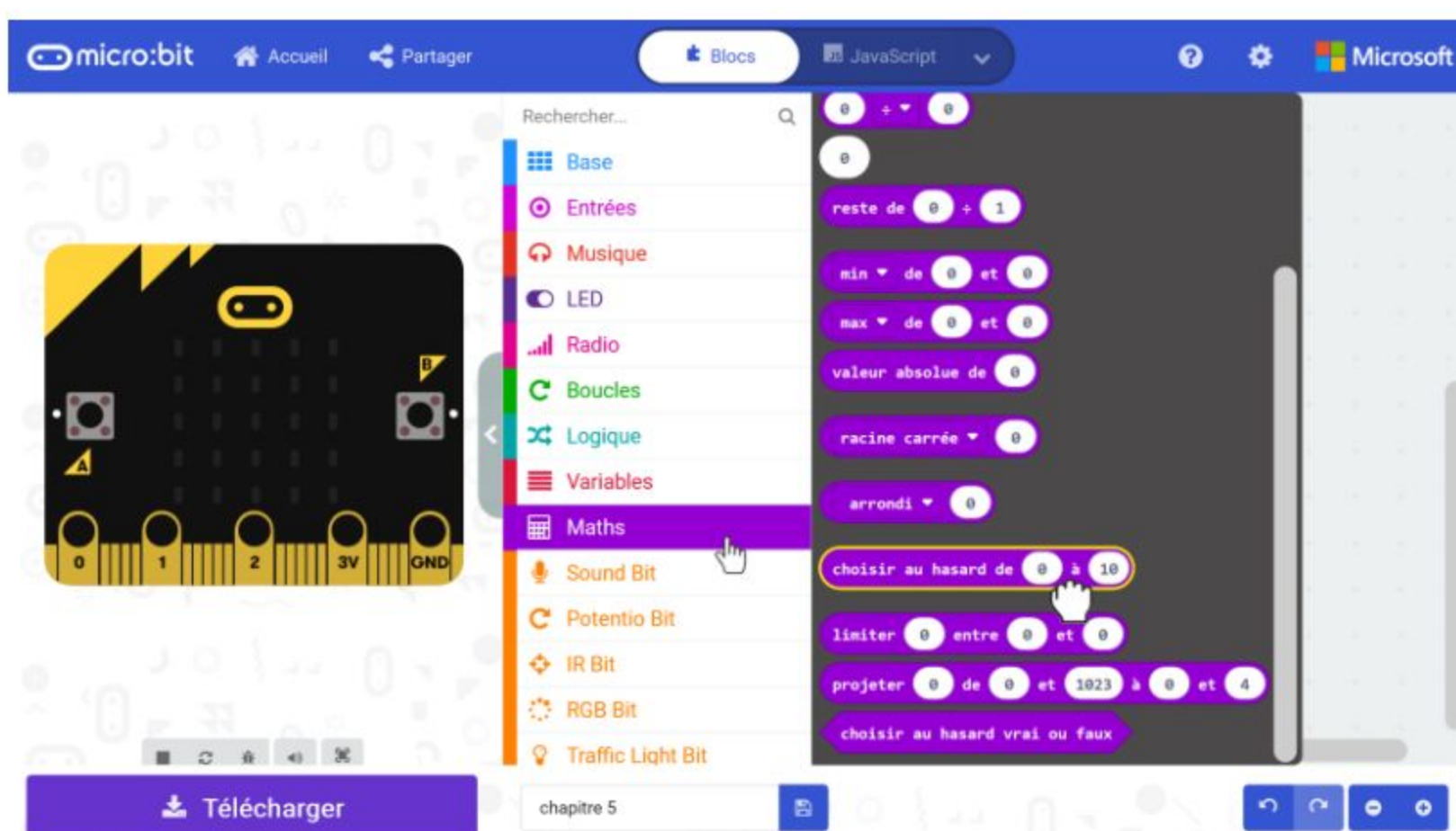
Étape 4 Clique sur la catégorie [**Musique**] et sélectionne le bloc [**démarrer la mélodie _répétition_**]. Remplace la mélodie “dadadum” par “ba ding”.



Étape 5 Clique sur la catégorie [**Base**] et sélectionne le bloc [**montrer nombre**].

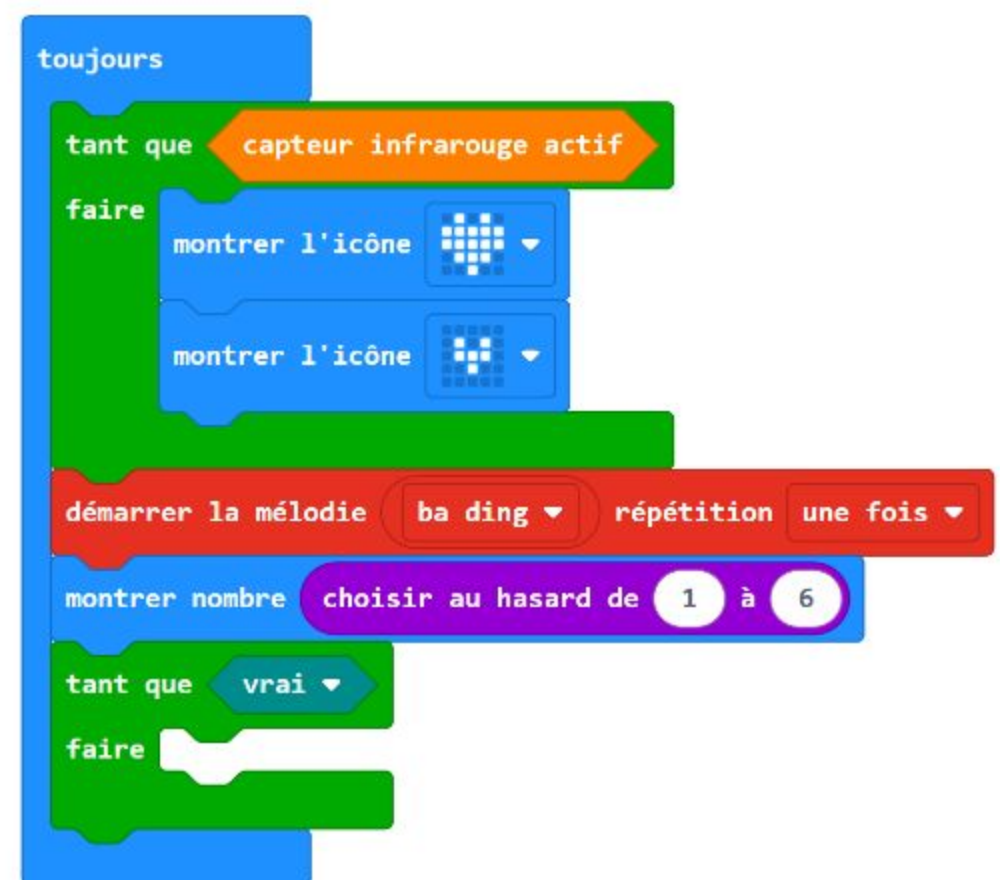
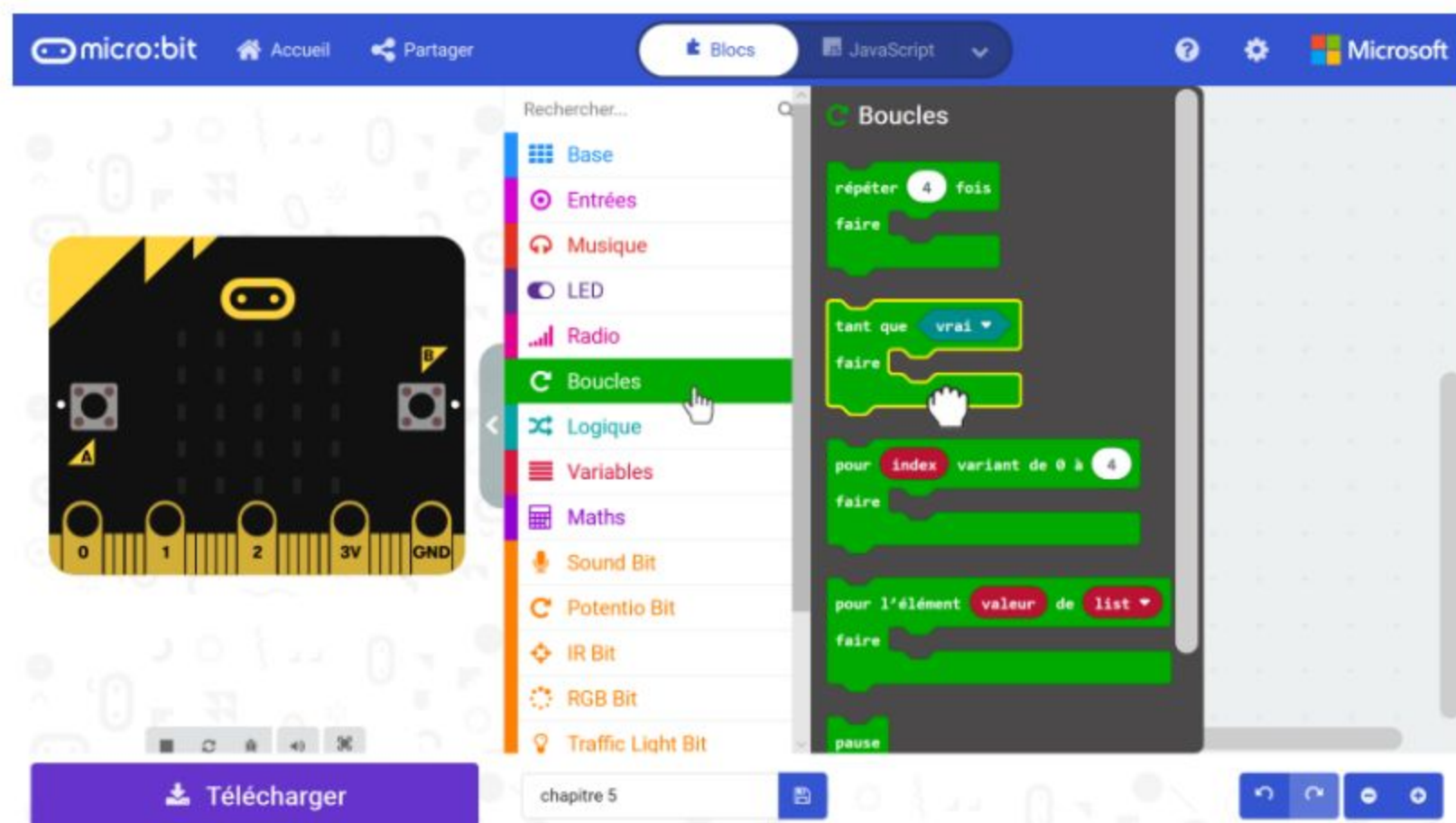


Étape 6 Clique sur la catégorie [**Maths**] et sélectionne le bloc [**choisir au hasard de _ à _**]. Change les bornes en 1 et 6.

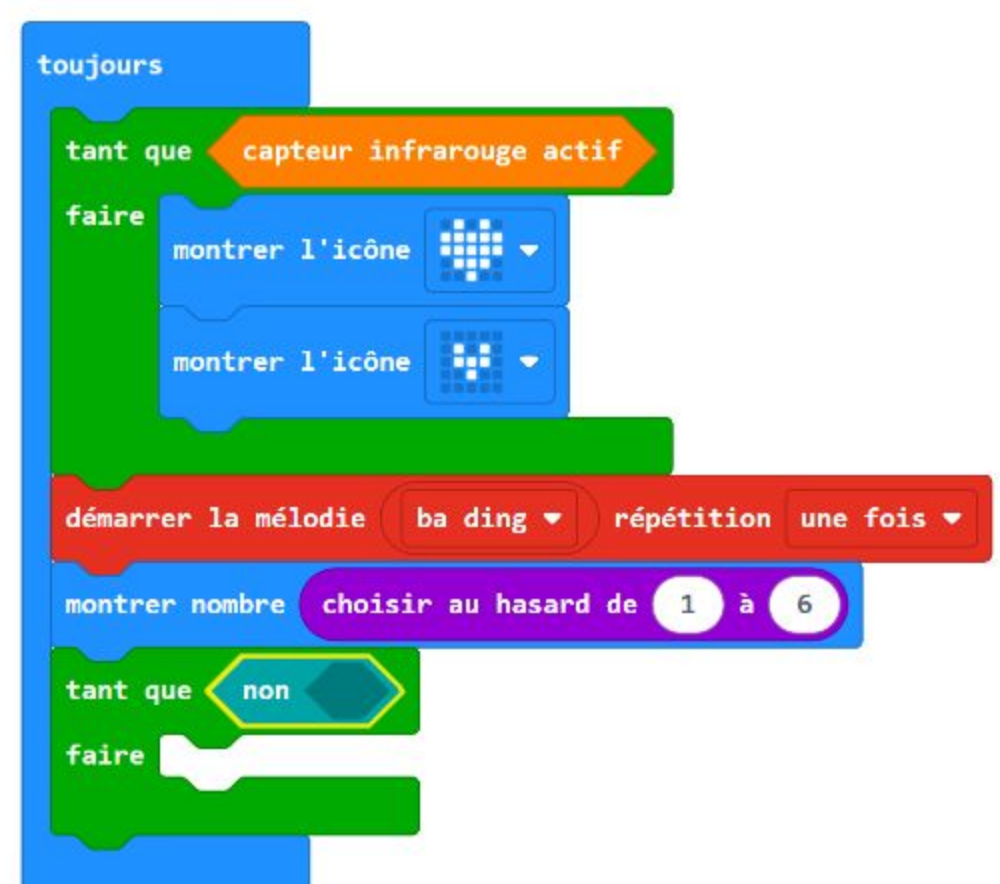
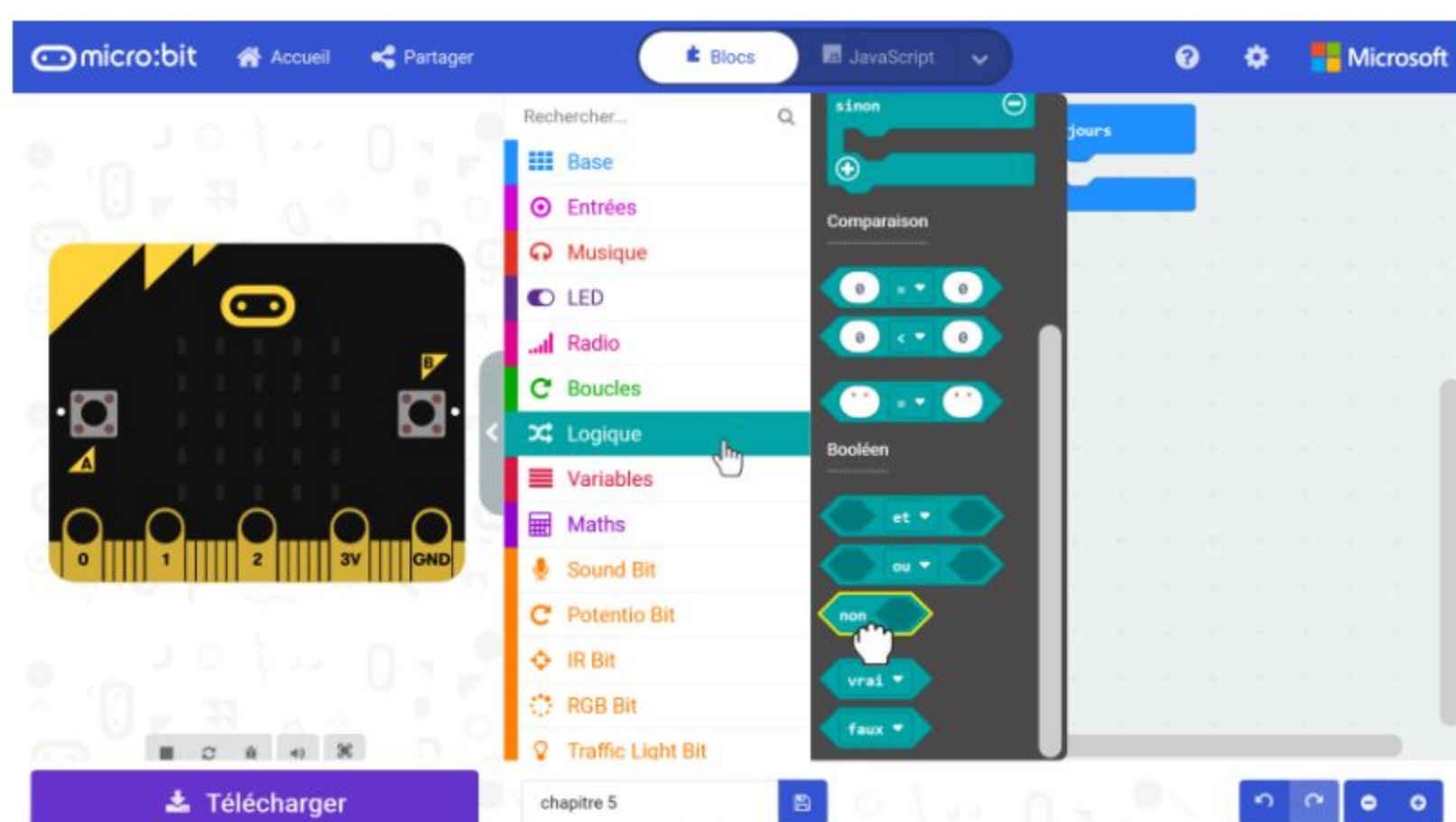


CHAPITRE 5 : Dé Numérique... Roule~

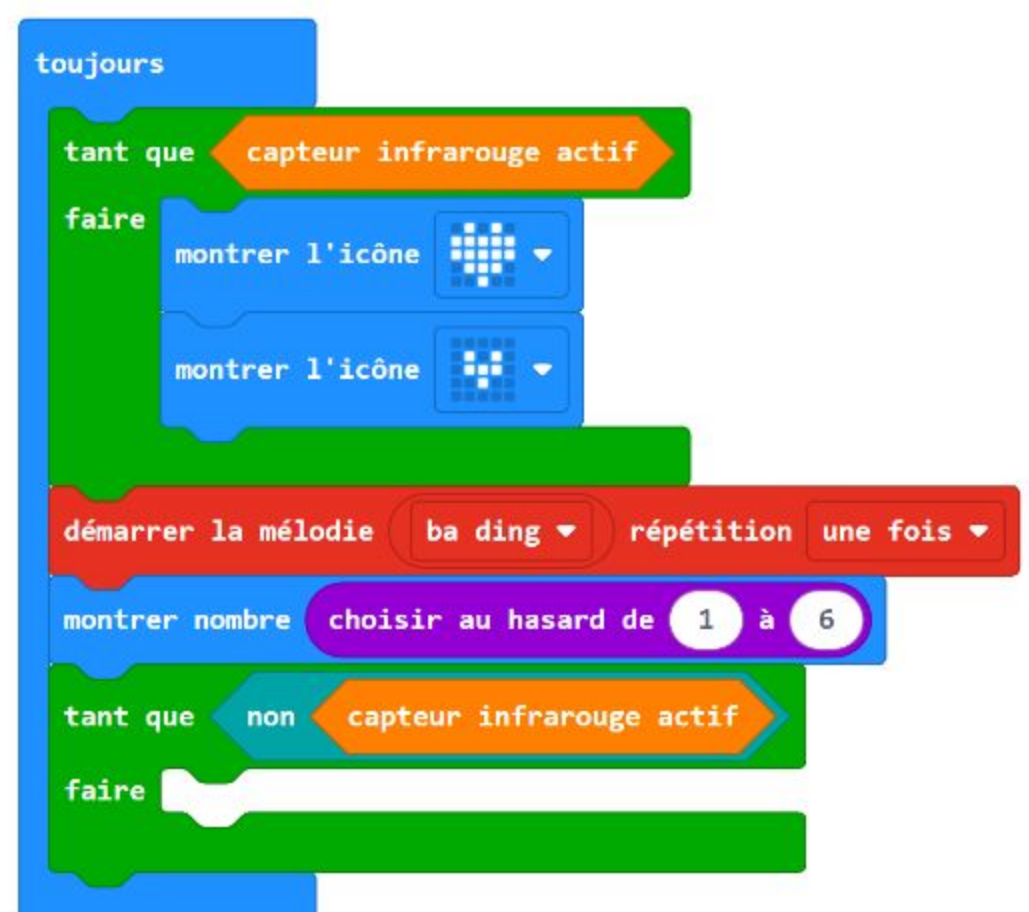
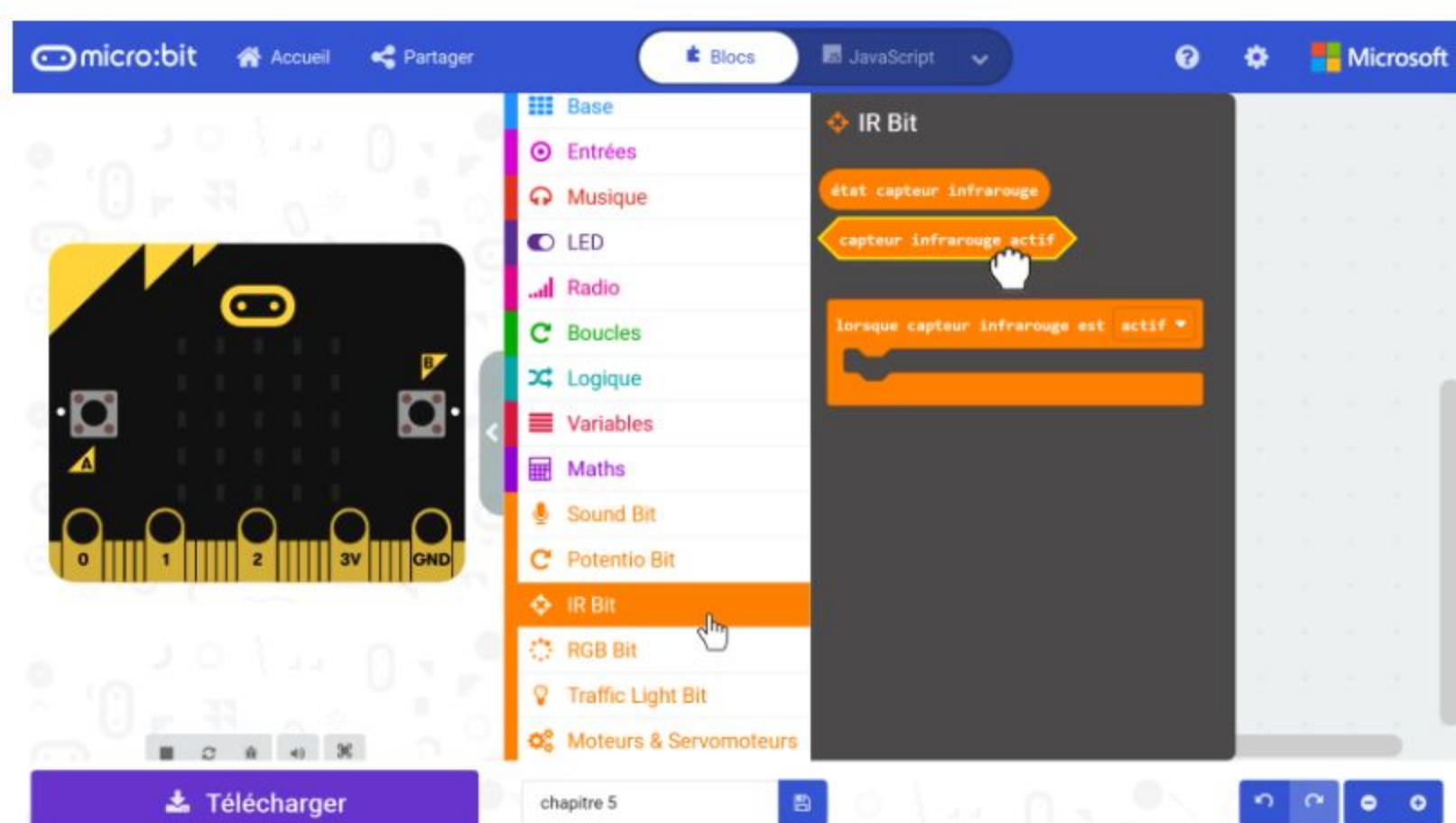
Étape 7 Clique sur la catégorie [**Boucles**] et sélectionne le bloc [**tant que _ faire**].



Étape 8 Clique sur la catégorie [**Logique**] et sélectionne le bloc [**non _**]. Insère-le dans la condition du bloc [**tant que _ faire**].



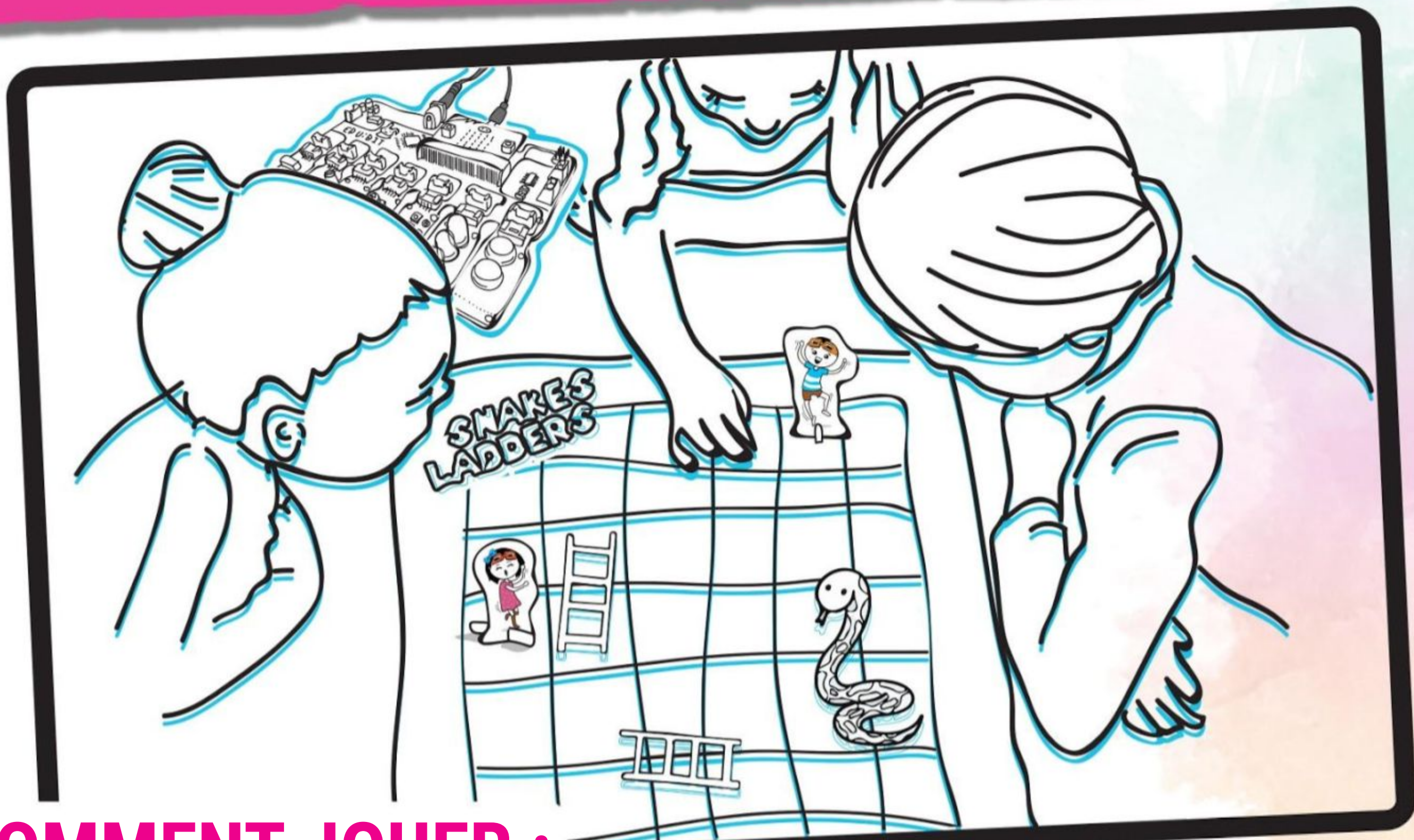
Étape 9 Clique sur la catégorie [**IR Bit**] et sélectionne le bloc [**capteur infrarouge actif**]. Insère-le dans le bloc [**non _**].



Étape 10 Flashe le code sur EDU:BIT.

Jouons

Serpents et échelles



COMMENT JOUER :

Chacun des joueurs choisit un meeples et le place sur la case départ "Start Here".

Les joueurs doivent ensuite "lancer un dé" à tour de rôle, en plaçant la paume de la main au dessus du capteur infrarouge. Dès que le cœur animé est affiché sur la carte, retirer la main.

Déplacer le meeples du nombre de cases indiqué sur la matrice de LEDs (entre 1 et 6).

Si le meeples atterrit en bas d'une échelle, il doit grimper tout en haut. S'il atterrit sur la tête d'un serpent, il glisse jusqu'au bout de sa queue.

Le premier joueur à atteindre la case 100 a gagné. Amuse-toi !



NOTE!

Le plateau de jeu et les pièces personnages (meeples) sont présents dans la boîte. Dépunche les personnages et les pieds, et assemble-les comme montré ci-dessus.

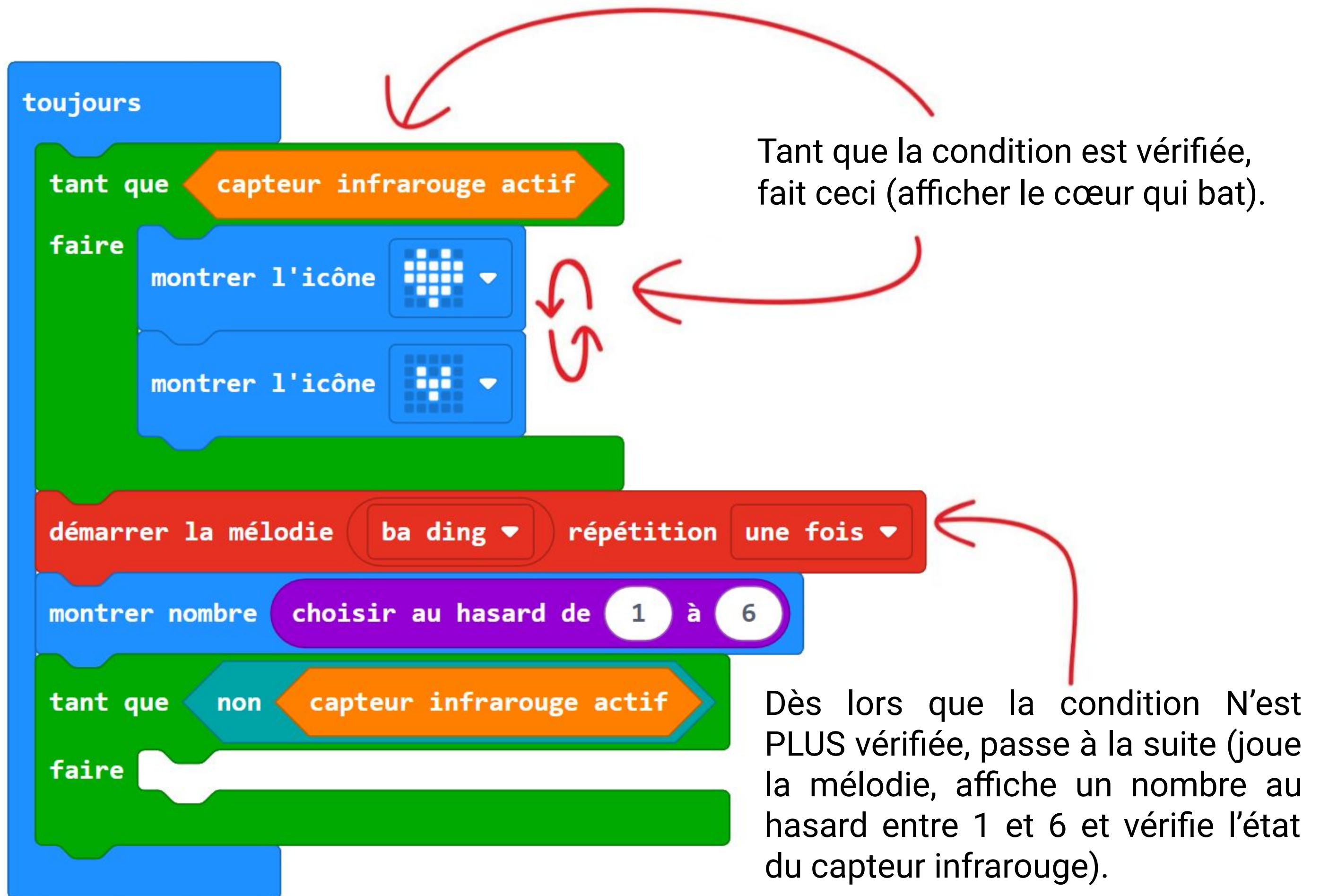
DÉCHIFFRE

LE CODE



Dans notre code précédent, nous avons utilisé un bloc [tant que _ faire] de la catégorie [Boucles]. Sais-tu comment une boucle fonctionne ?

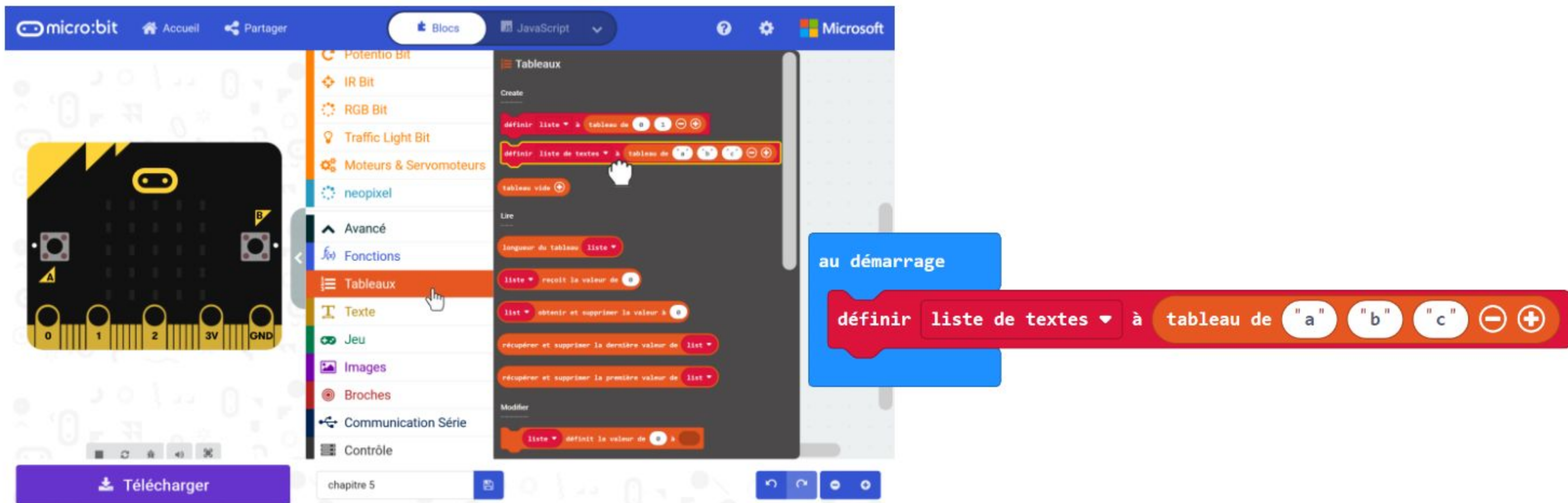
Lorsque le programme arrive au bloc [tant que _ faire], il vérifie la condition. Si la condition est vérifiée (ie : est VRAIE), le programme exécutera le(s) bloc(s) imbriqué(s) dans le bloc [tant que _ faire]. Il revient alors au début du bloc [tant que _ faire], teste à nouveau la condition, etc... Il continue tant que la condition est vérifiée. Mais une fois que la condition n'est plus vérifiée (ie : est FAUSSE), le programme sort de la boucle et exécute le bloc suivant.



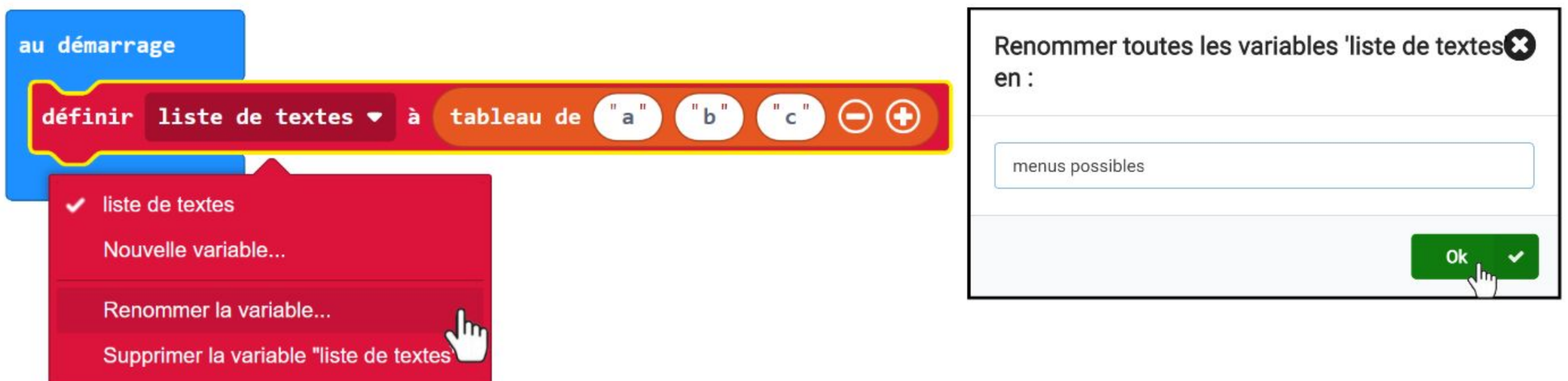


En plus d'utiliser EDU:BIT comme un simulateur de dé numérique, tu peux aussi modifier le programme précédent afin qu'EDU:BIT t'aide à faire un choix lorsque plusieurs options s'offrent à toi et que tu n'arrives pas à te décider. Par exemple, que va-t-on manger ce midi ? Miam miam~

Étape 11 Clique sur la catégorie [**Avancé**] : [**Tableaux**] et sélectionne le bloc [**définir liste de textes à tableau de _**]. Imbrique-le dans le bloc [**Base**] : [**au démarrage**].



Étape 12 Clique sur le bouton [**liste de textes**] et sélectionne "Renommer la variable...". Saisis "menus possibles" dans la fenêtre pop-up et clique sur OK.



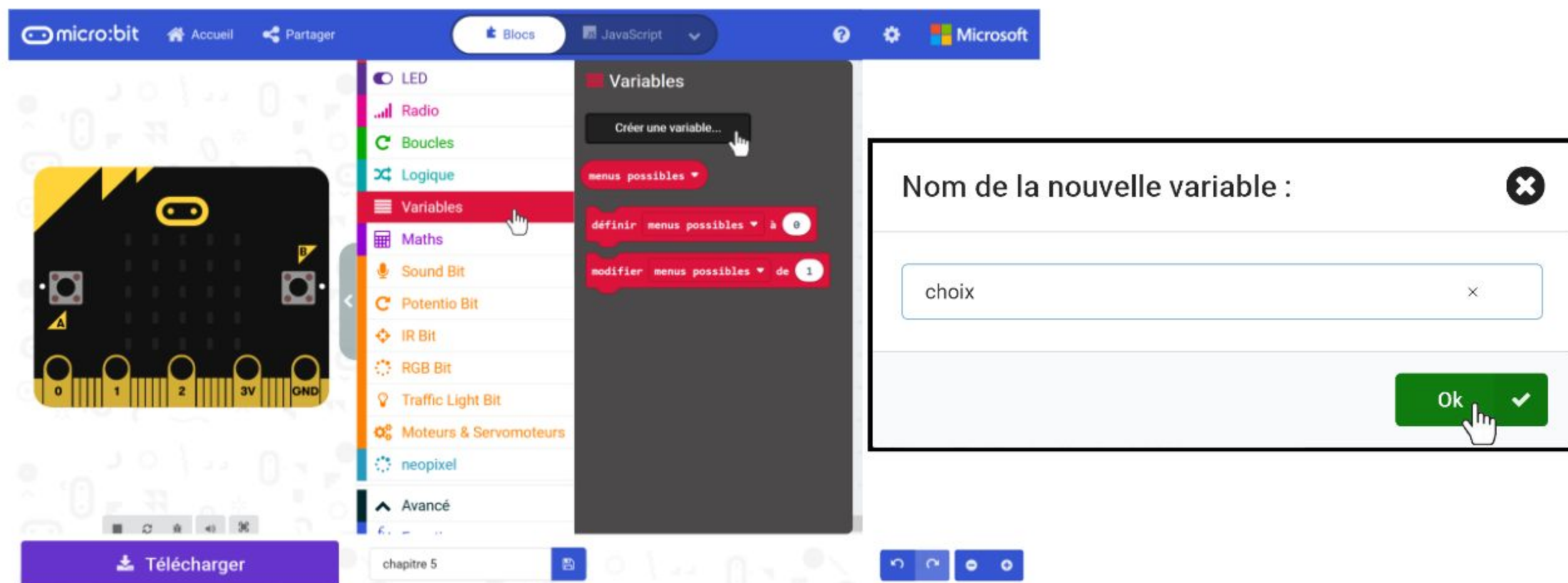
Étape 13 Complète alors les cases du tableau avec les menus de ton choix.



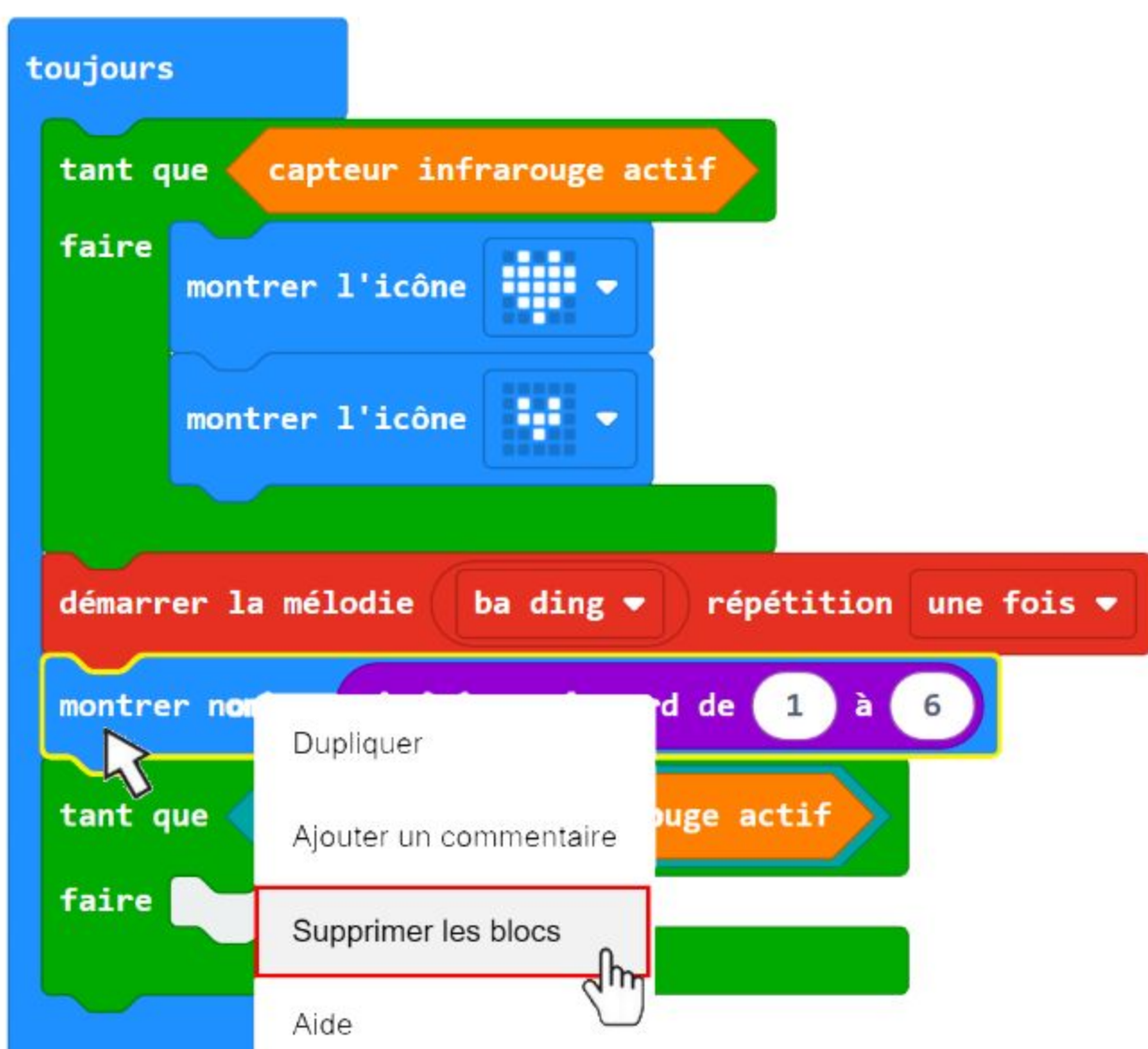
Clique sur ce bouton si tu veux ajouter des éléments.

CHAPITRE 5 : Dé Numérique... Roule~

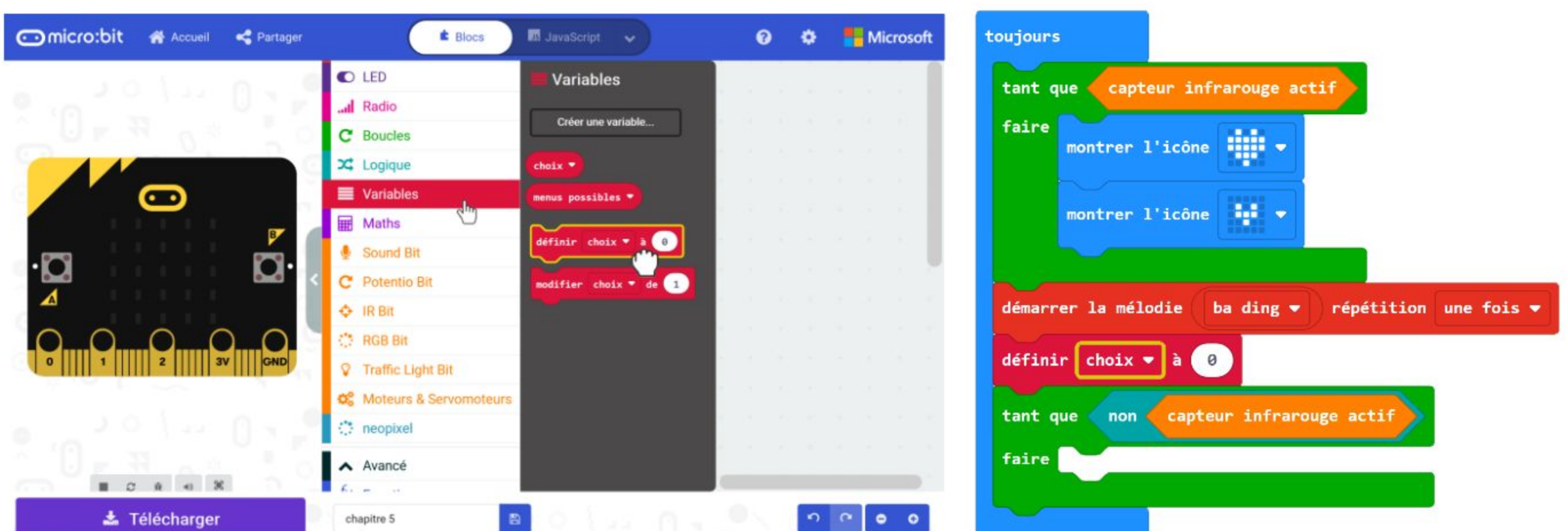
Étape 14 Clique sur la catégorie [**Variables**] et crée une nouvelle variable nommée "choix".



Étape 15 Fais un clic droit sur le bloc [**montrer nombre [choisir au hasard de _ à _]**] et sélectionne "Supprimer les blocs".

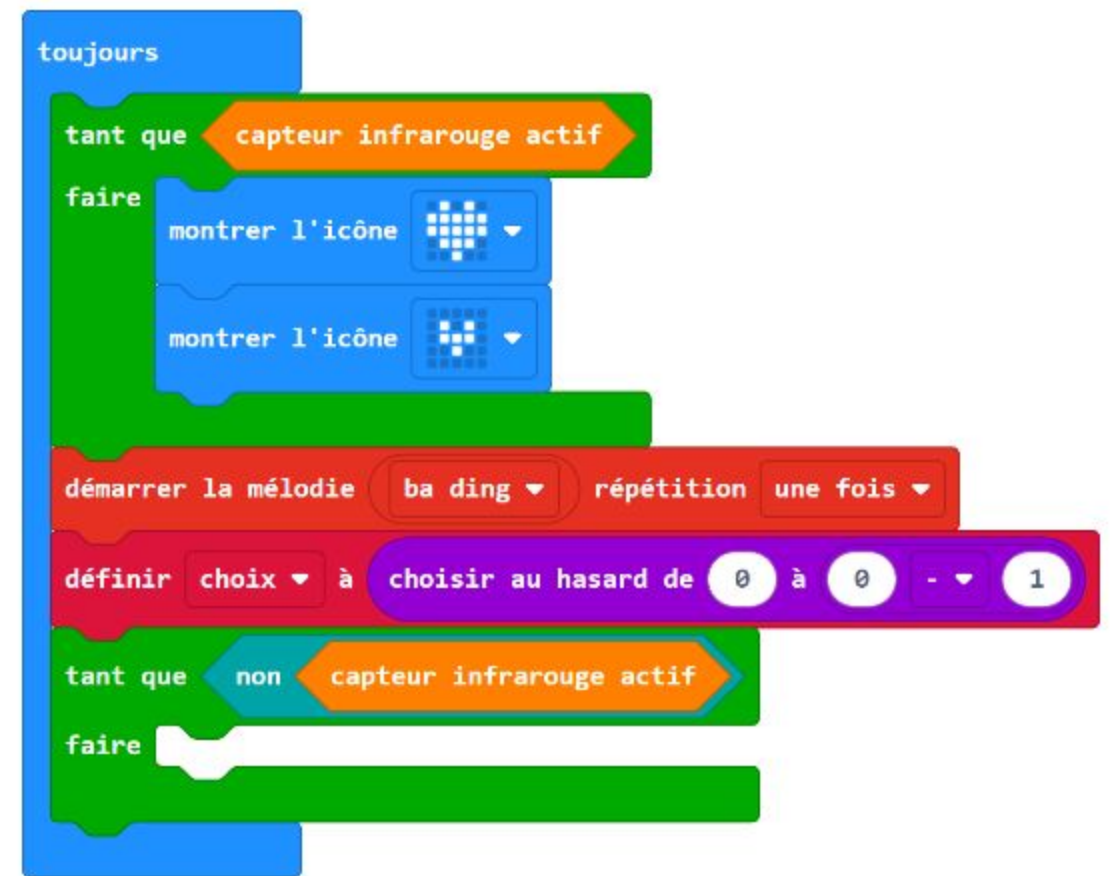


Étape 16 Clique sur la catégorie [**Variables**] et sélectionne [**définir _ à _**]. Intercalle-le entre les blocs [**démarrer la mélodie _ répétition _**] et [**tant que _ faire**]. Sélectionne la variable "choix".

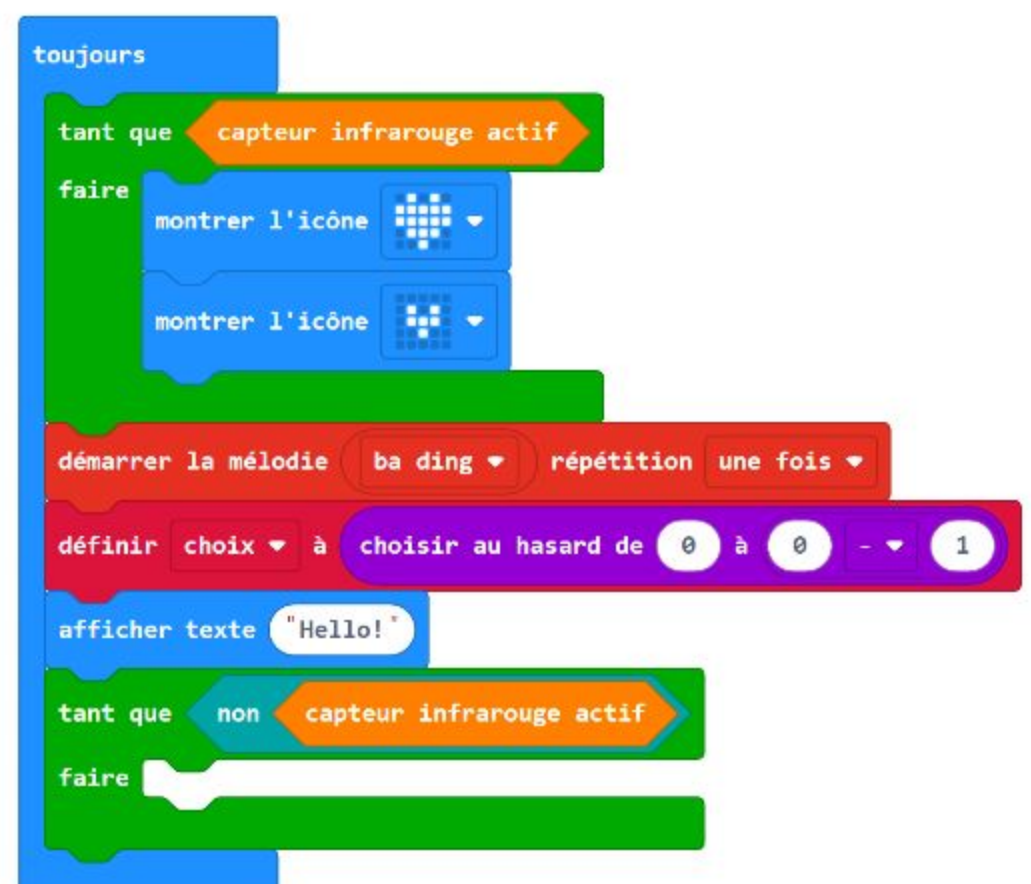
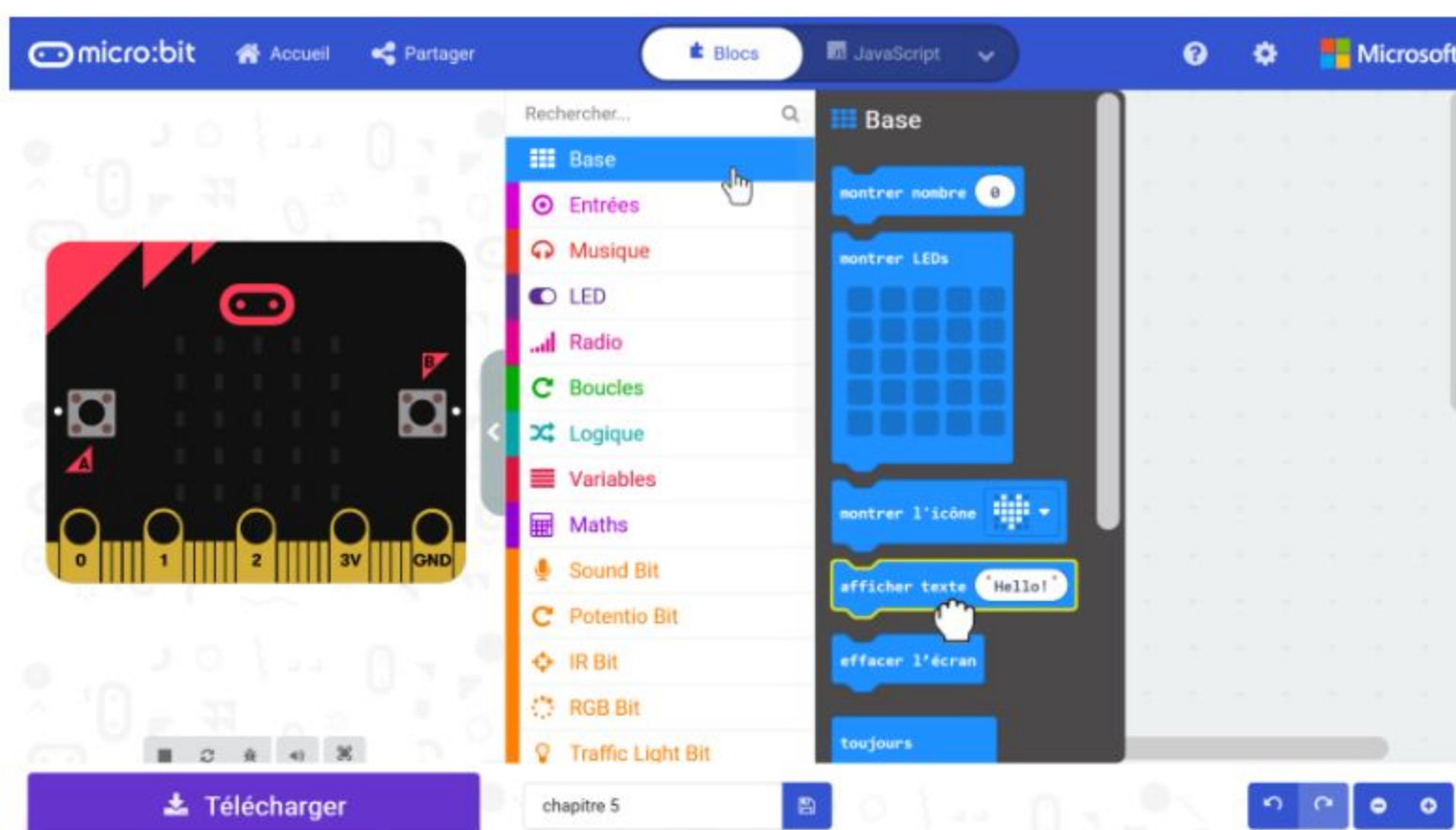




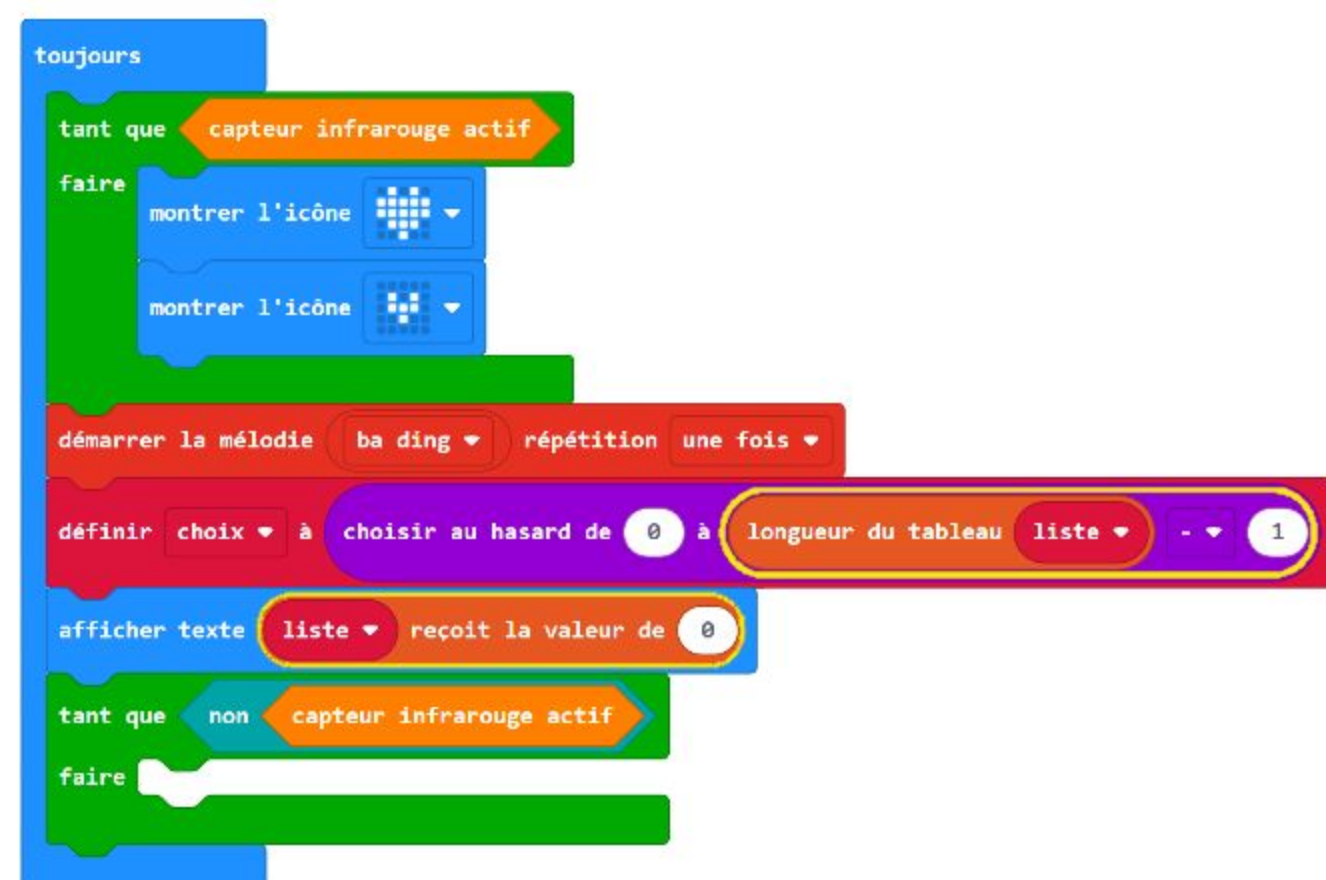
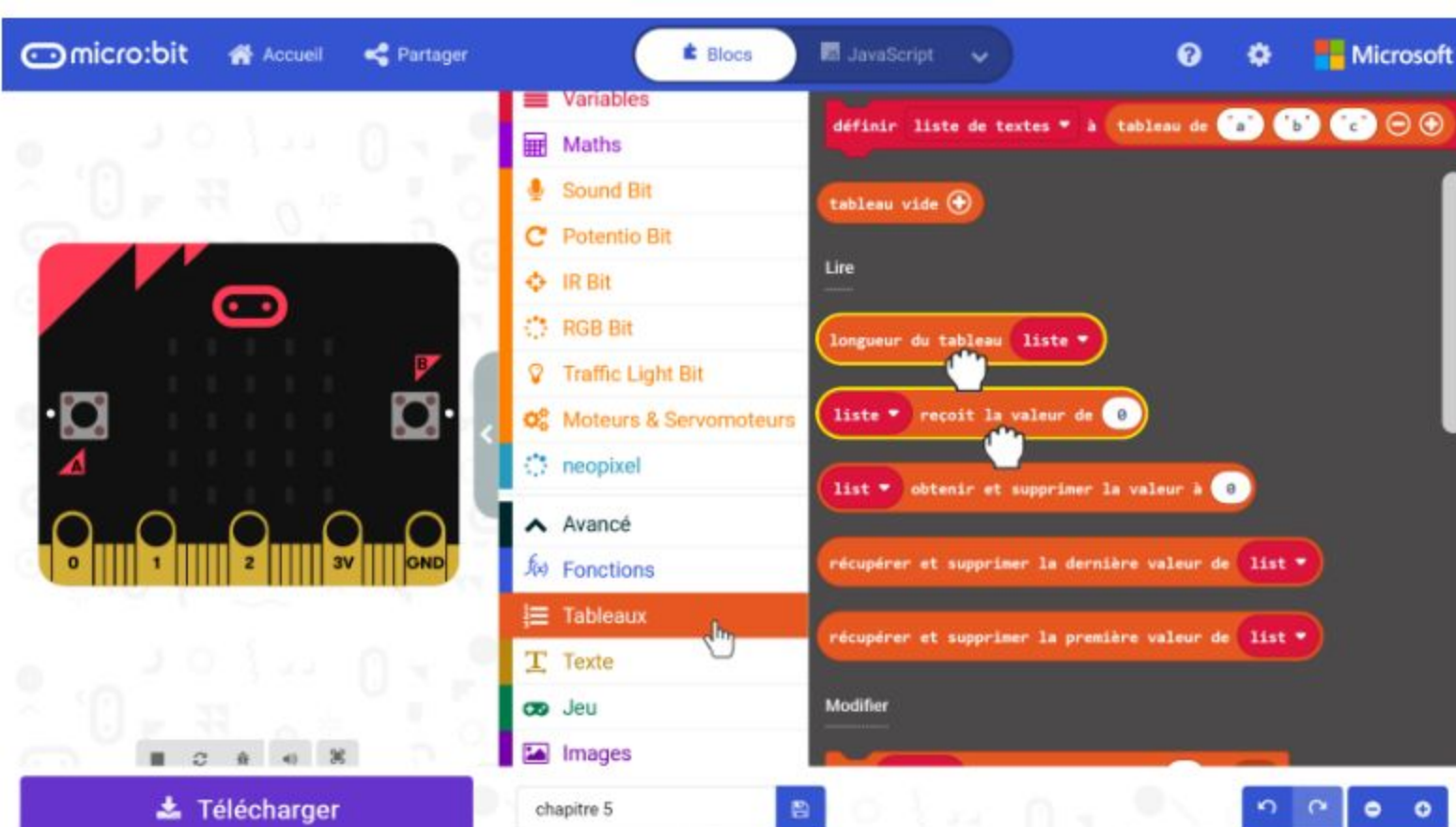
Étape 17 Clique sur la catégorie [Maths] et ajoute le bloc [choisir au hasard de _ à _] à ton code. Dans la borne supérieure, insères-y un bloc [_ - _], puis écris 1 dans le deuxième terme de la différence.



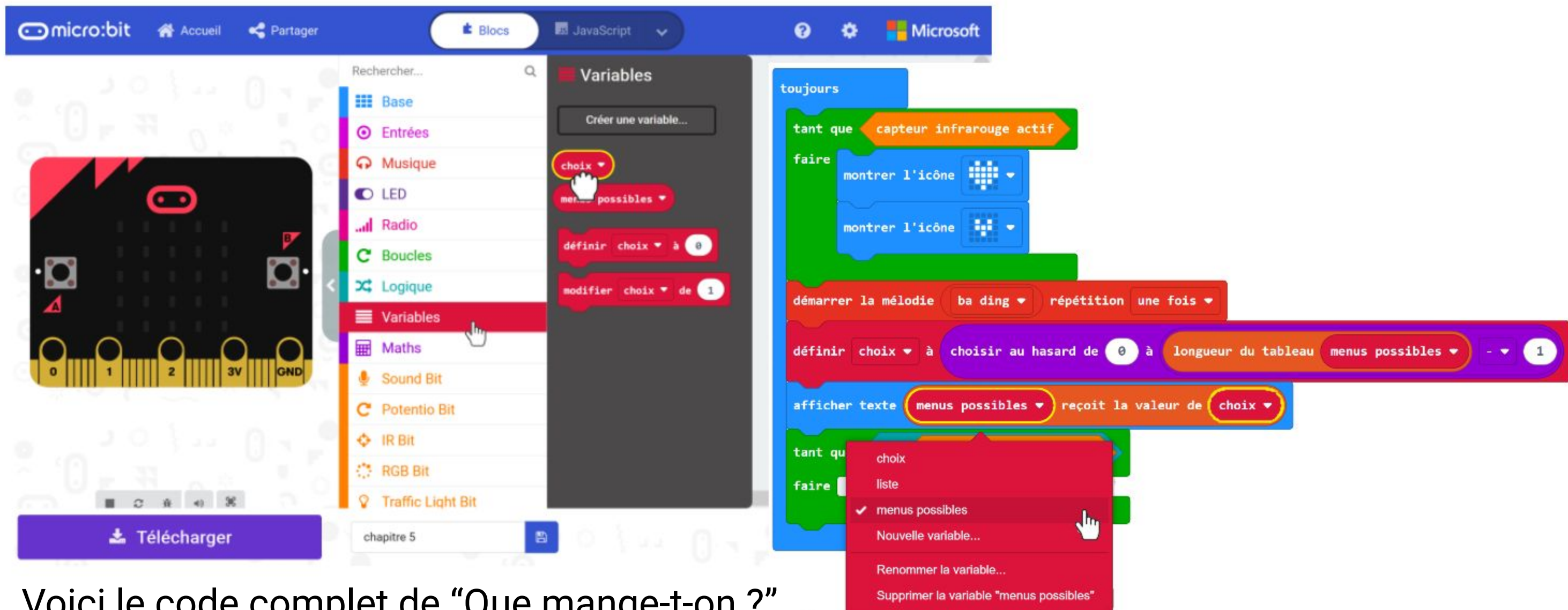
Étape 18 Clique sur la catégorie [Base] et sélectionne le bloc [afficher texte].



Étape 19 Clique sur la catégorie [Avancé] : [Tableaux] et ajoute les blocs [longueur du tableau _] et [_ reçoit la valeur de _] comme ci-dessous.



Étape 20 Clique sur le bouton [liste] et sélectionne la variable "menus possibles" pour chacun des deux blocs. Enfin, clique sur la catégorie [Variables] et sélectionne le bloc [choix]. Imbrique-le dans l'emplacement vide du bloc [_ reçoit la valeur de _].



Voici le code complet de "Que mange-t-on ?".

```
au démarrage
  définir menus possibles à tableau de "Pizza" "Spaghettis" "Burger"

toujours
  tant que capteur infrarouge actif
  faire
    montrer l'icône
    montrer l'icône

  démarrer la mélodie ba ding répétition une fois

  définir choix à choisir au hasard de 0 à longueur du tableau menus possibles - 1

  afficher texte menus possibles reçoit la valeur de choix

  tant que non capteur infrarouge actif
  faire
```

Ce ne sont que des exemples. Tu peux les remplacer par d'autres plats, et même en ajouter d'autres si tu veux.



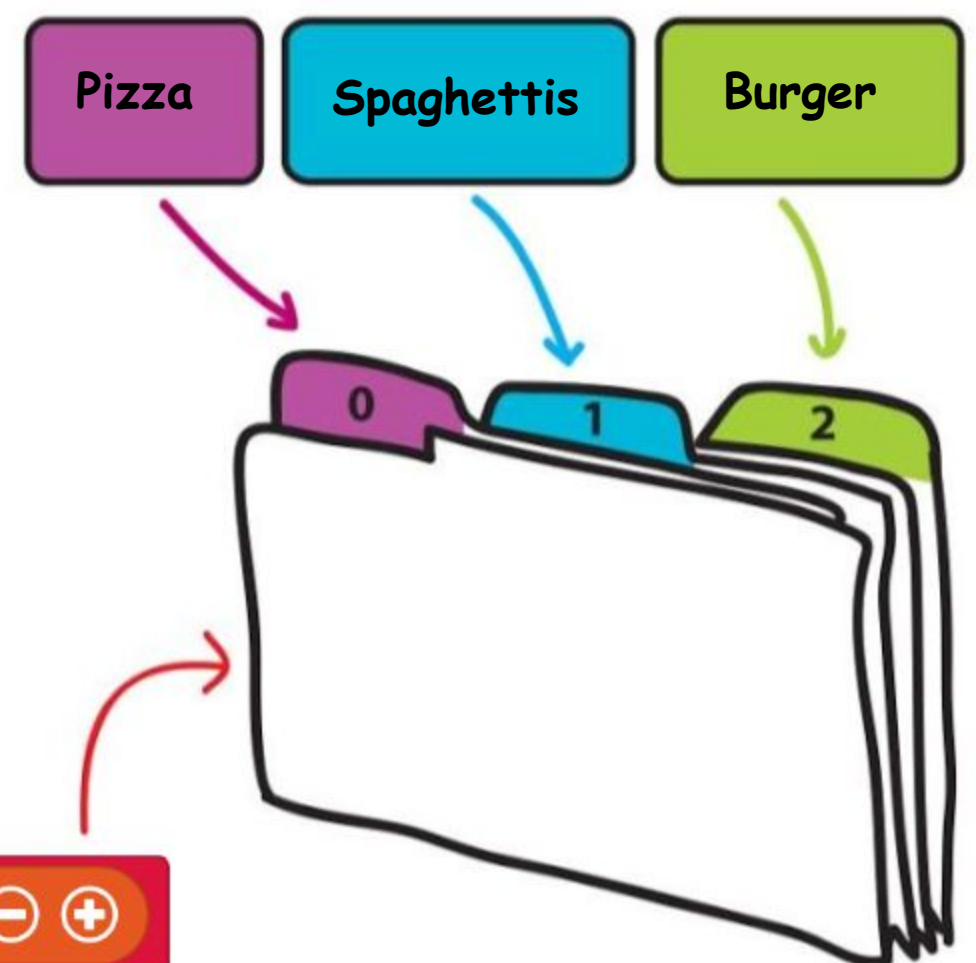
La prochaine fois que tu ne sais pas quoi manger, tu peux utiliser EDU:BIT pour décider à ta place en déclenchant le capteur infrarouge avec ta main. Tu peux aussi modifier le code pour t'aider à choisir un jeu auquel tu peux jouer avec tes amis. Sais-tu quelles modifications il faut apporter au code pour cela ?

DÉCHIFFRE

LE CODE

Un **tableau** est une liste ou collection d'objets associés à des variables. Tu peux te le représenter comme un classeur ou un trieur ayant plusieurs rubriques, chaque rubrique pouvant stocker une information. En utilisant des tableaux on peut donc facilement modifier notre code pour ajouter ou supprimer des éléments d'une liste.

Dans ce code par exemple, nous avons créé un tableau nommé **"menus possibles"** contenant trois éléments. Nous pouvons ainsi facilement modifier les plats inscrits dans chaque élément. Nous pouvons aussi augmenter ou réduire le nombre d'éléments contenus dans la liste en cliquant simplement sur les boutons **+** et **-**.



au démarrage

Indices de la liste 0 1 2

définir menus possibles à tableau de "Pizza" "Spaghettis" "Burger" - +

toujours

tant que capteur infrarouge actif

faire

montrer l'icône

montrer l'icône

démarrer la mélodie ba ding répétition une fois

définir choix à choisir au hasard de 0 à longueur du tableau menus possibles - 1

afficher texte menus possibles reçoit la valeur de choix

tant que non capteur infrarouge actif

faire

Dès que la condition n'est plus vérifiée, EDU:BIT passe à la suite :

(i) joue la mélodie "ba ding" une fois,

(ii) choisit au hasard un élément du tableau qui deviendra le "choix", et

(iii) l'affiche sur l'écran de LEDs.

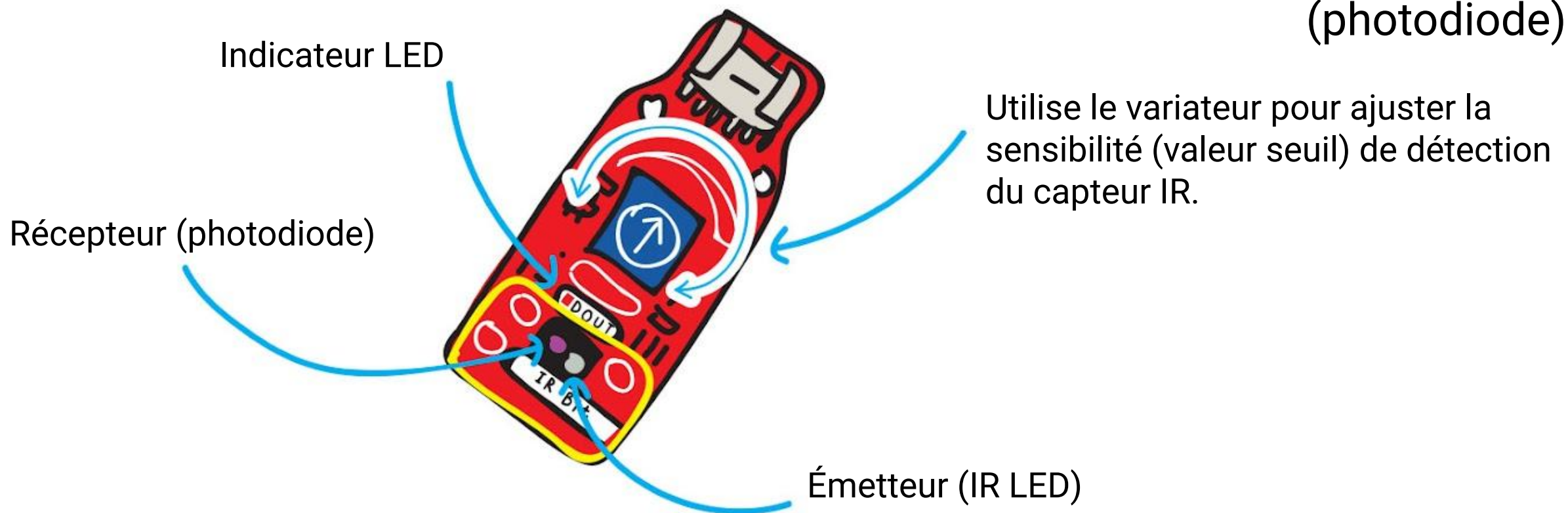
En programmation, on commence à compter à partir de 0, et non 1. C'est pourquoi "Pizza" est à l'indice 0 de la liste et que le dernier élément "Burger" est à l'indice 2, bien qu'il s'agisse du troisième élément.



FAIT AMUSANT !



Un **capteur infrarouge (IR)** est un composant électronique couramment utilisé pour détecter les objets. Un capteur IR est constitué de deux parties : un émetteur (LED IR) et un récepteur (photodiode).



Comment ça marche ?

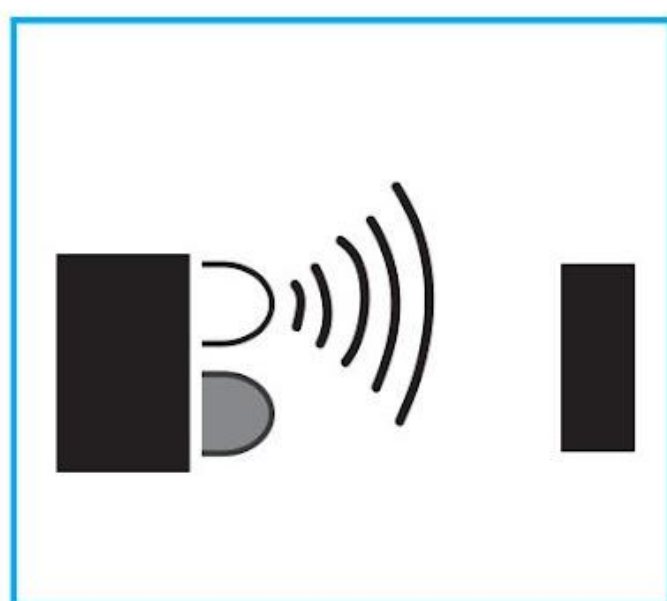
La LED infrarouge (LED IR) émet une lumière infrarouge qui se reflète dans le récepteur si un objet est situé devant le capteur. L'IR Bit est déclenché si la quantité de lumière réfléctée est supérieure à une valeur seuil. Lorsqu'il est déclenché, un indicateur LED sur l'IR Bit clignote.

S'il n'y a aucun objet ou que l'objet est trop éloigné, alors une trop faible quantité, voire aucune lumière infrarouge ne sera réfléctée sur le récepteur. Ainsi, IR Bit ne sera pas déclenché.

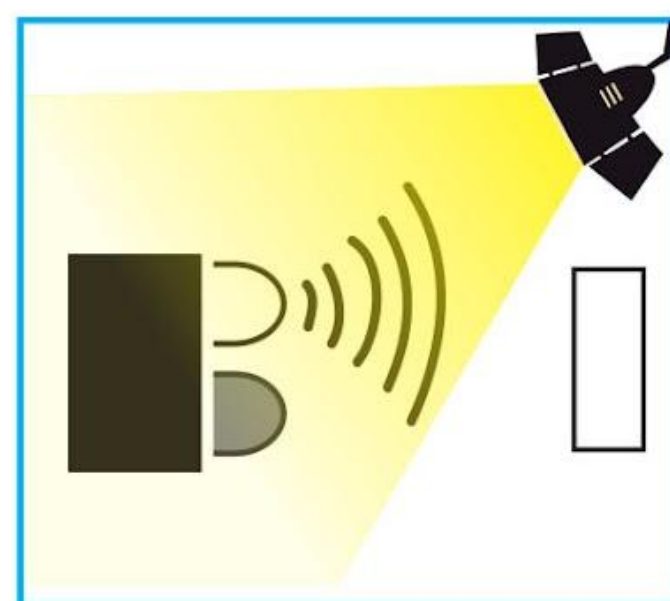
Par ailleurs, le capteur IR peut ne pas fonctionner comme prévu dans les conditions suivantes :



Objet trop petit



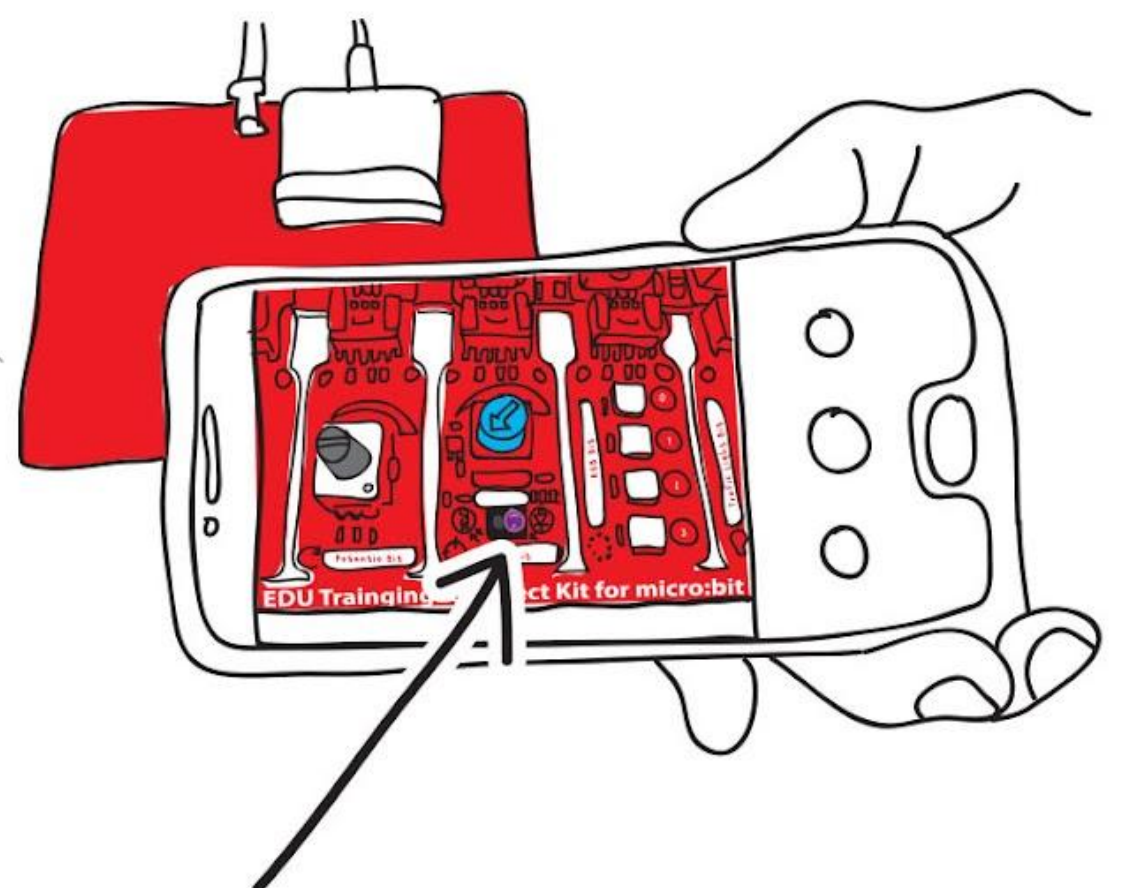
Objet noir ou de surface trop sombre



Interférences lumineuses

Le sais-tu ?

La lumière infrarouge est invisible à l'oeil nu. Cependant, tu peux la voir en l'observant simplement grâce au capteur photo d'un téléphone portable.

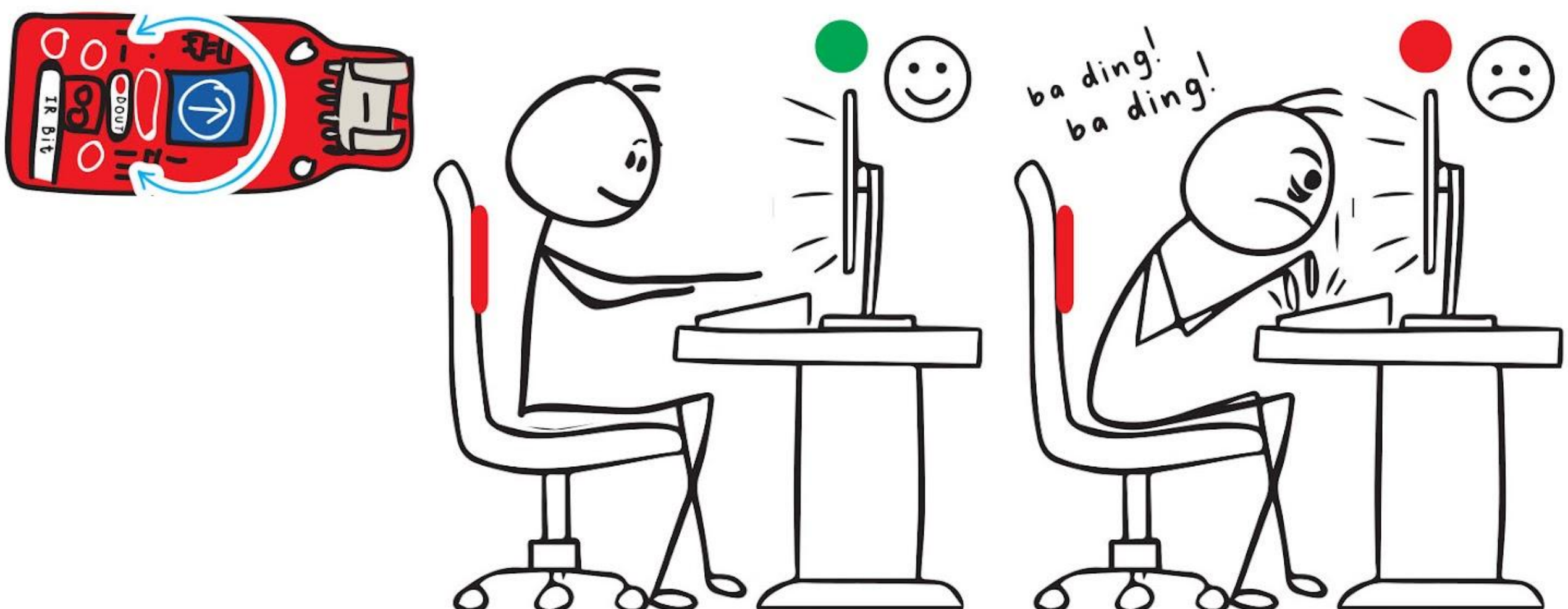


APPLICATION DÉFI

Programme EDU:BIT pour qu'il serve de capteur de position "anti-affalement" sur la chaise.	
au démarrage	Afficher un rappel "Tiens-toi droit !".
lorsque capteur infrarouge est actif	Afficher un sourire sur la matrice de LEDs et allumer la LED verte.
lorsque capteur infrarouge est inactif	Jouer la mélodie "ba ding" une fois. Afficher une grimace sur la matrice de LEDs et faire clignoter la LED rouge.

Comment ça marche ?

Fixe EDU:BIT sur le dossier de la chaise comme ci-dessous. Assieds-toi confortablement et correctement, le dos bien droit. Ajuste le potard bleu de IR Bit jusqu'à ce que l'indicateur LED s'allume (le capteur IR détecte ton dos). Ce procédé s'appelle un calibrage.



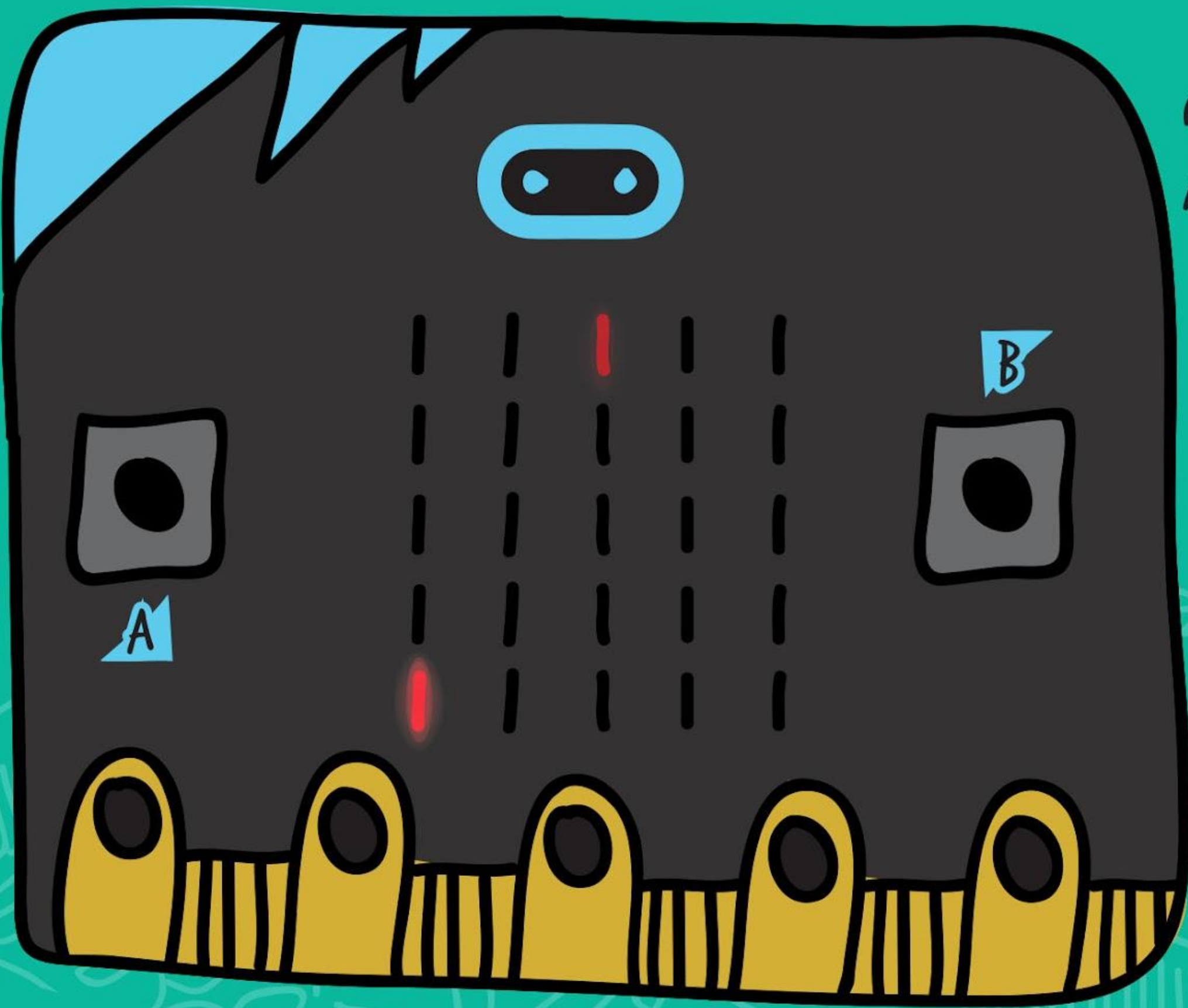
Tu devras recalibrer l'IR Bit si tu portes un vêtement d'une couleur différente. Sais-tu pourquoi ?



CHAPITRE 6

C'est Toi le Chat !

Potentio Bit



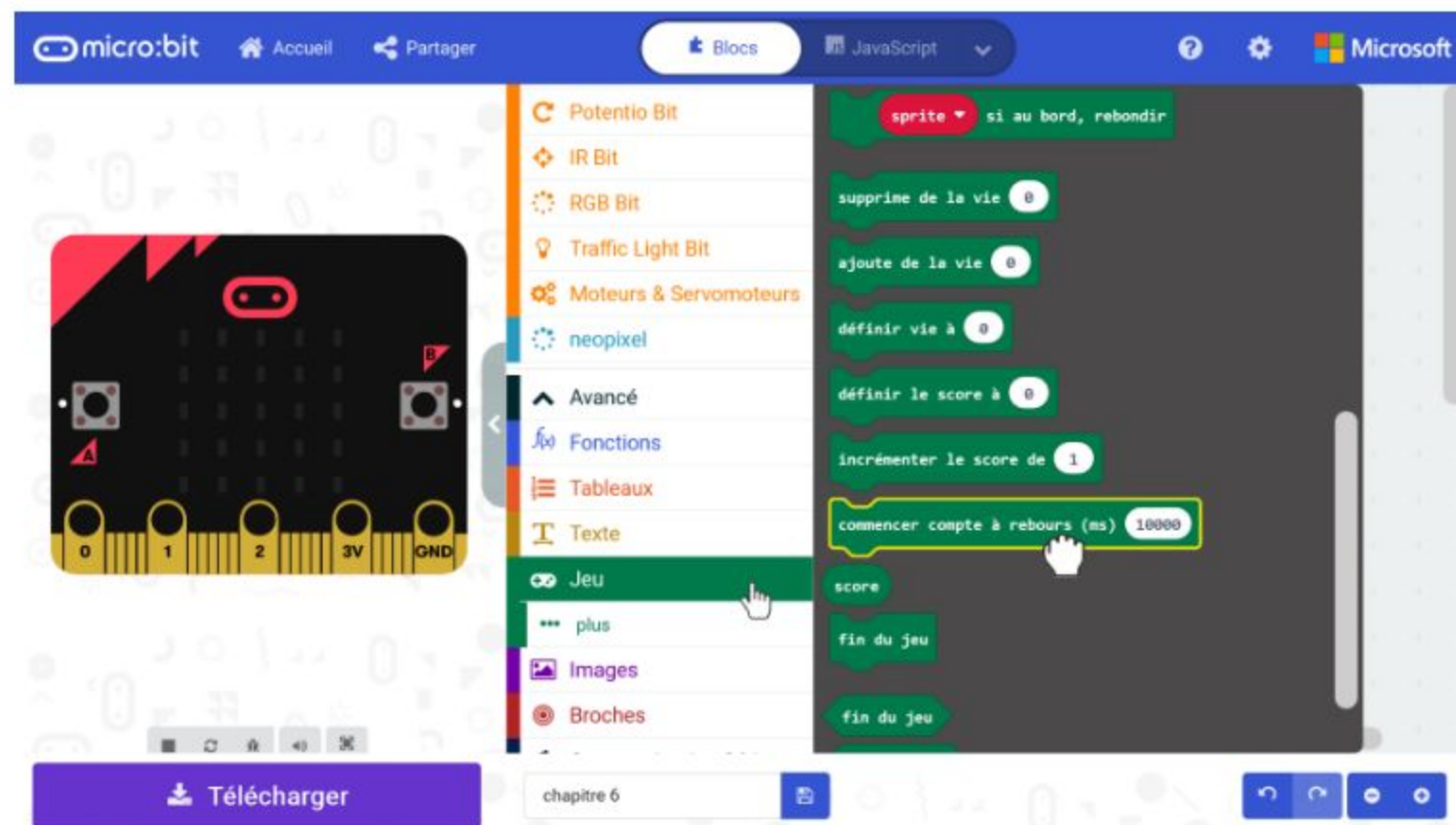
Scanne-moi !

link.cytron.io/edubit-chapter-6

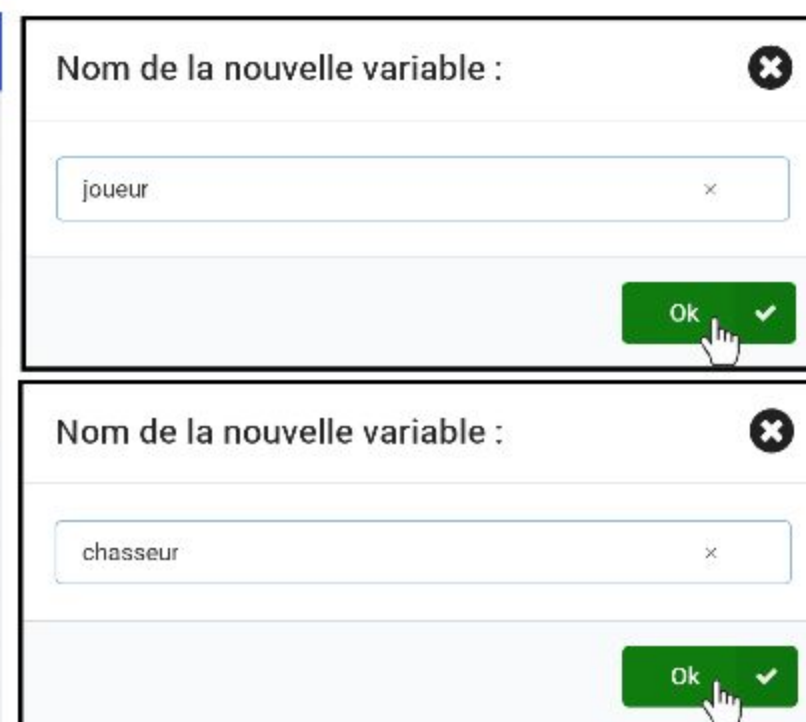
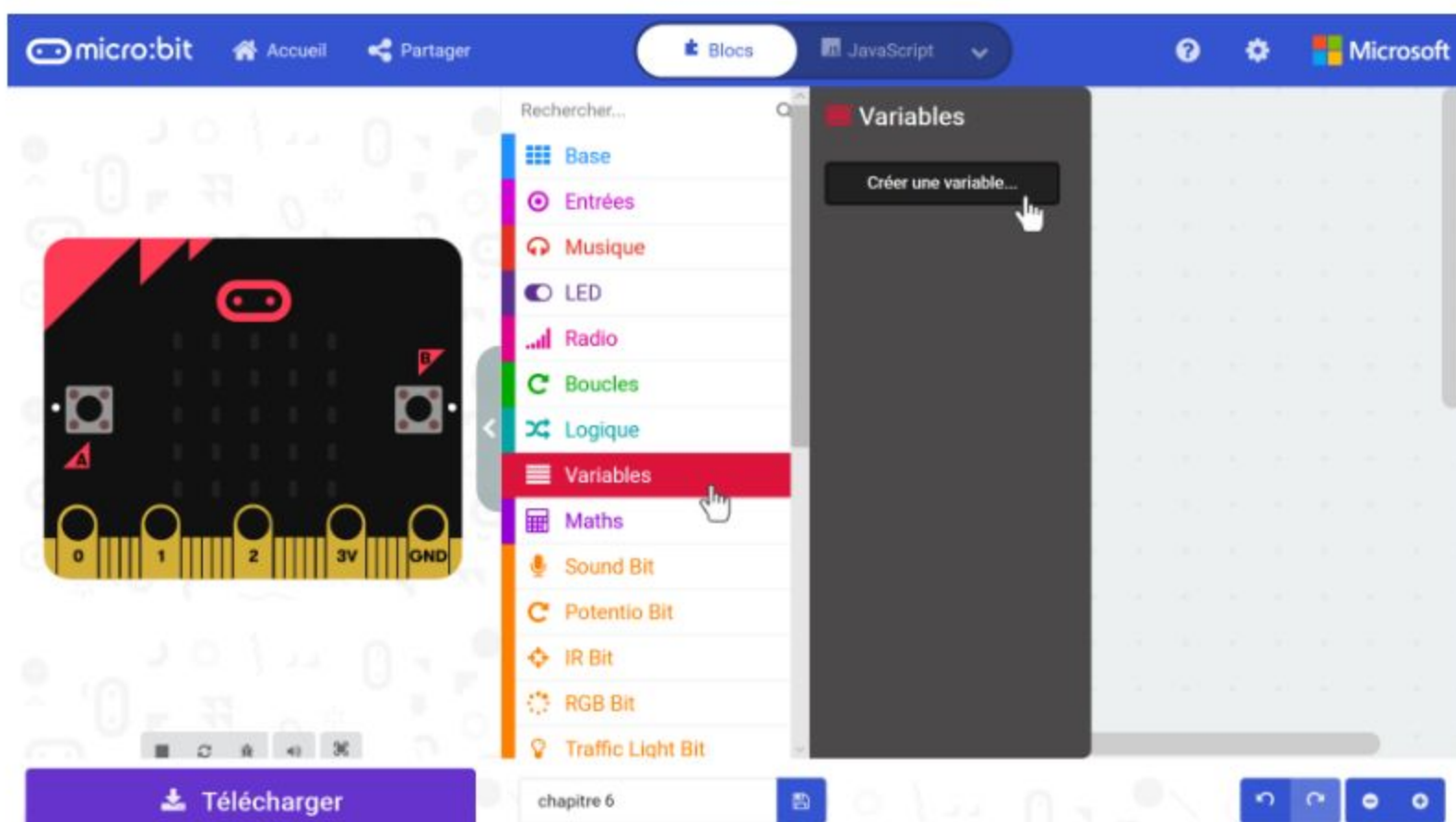


Codons !

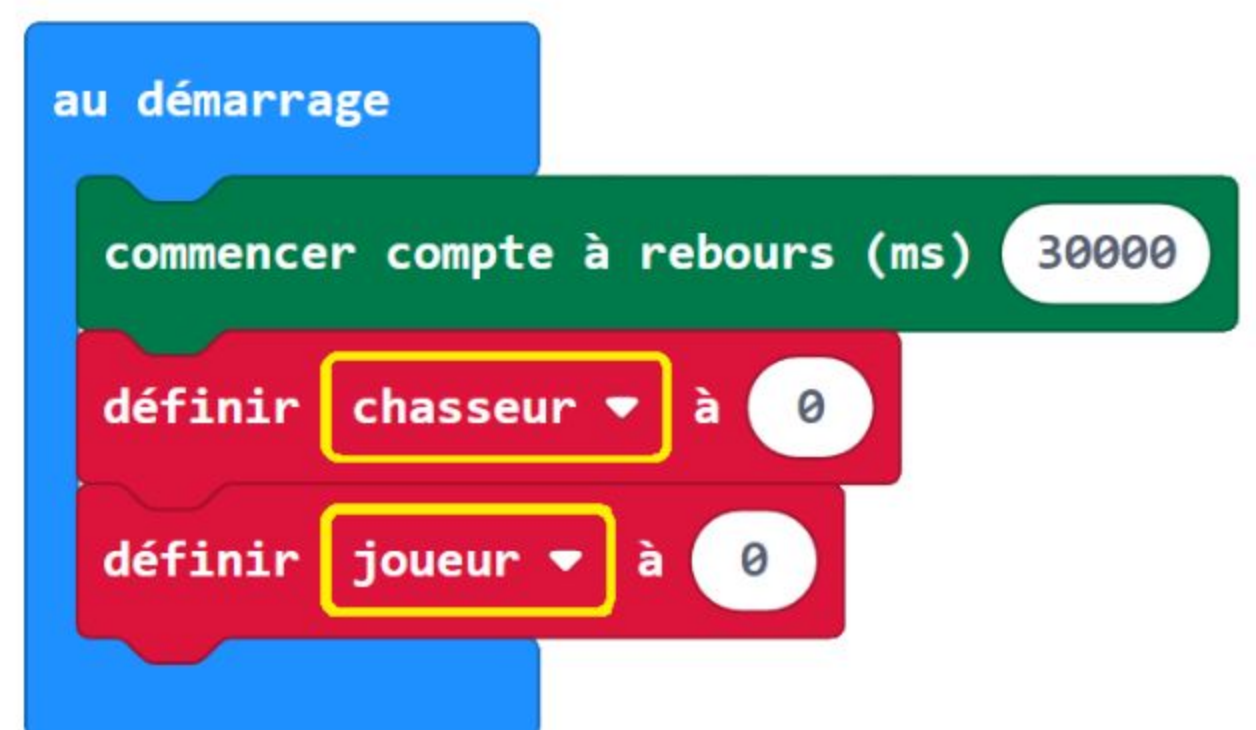
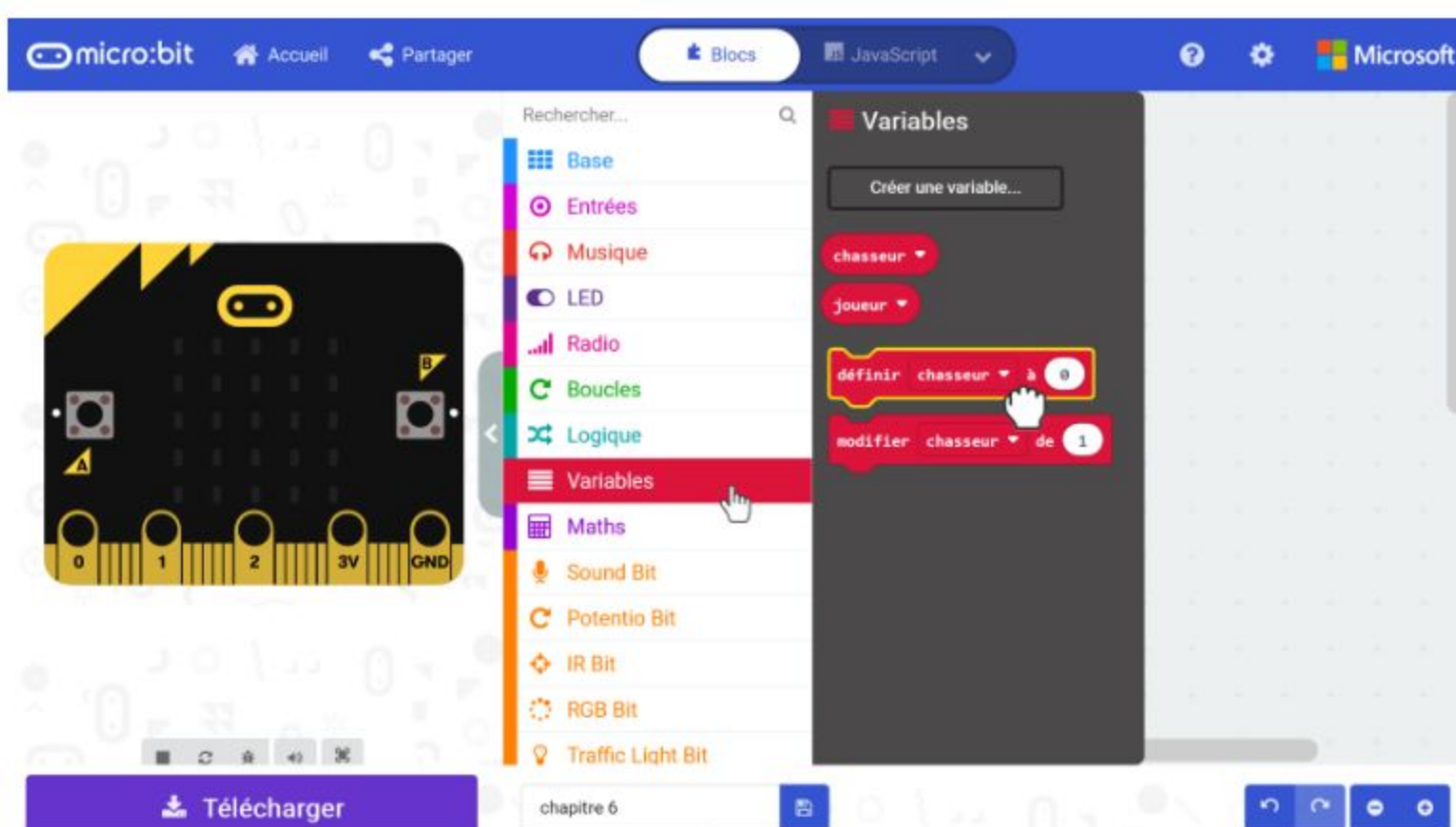
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT (voir page 40). Clique sur [**Avancé**] puis sur la catégorie [**Jeu**]. Sélectionne le bloc [**commencer compte à rebours (ms) _**], et imbrique-le dans le bloc [**au démarrage**]. Remplace la valeur par **30 000**.



Étape 2 Clique sur la catégorie [**Variables**], puis sur [**Créer une variable...**]. Saisis "joueur" dans la fenêtre pop-up puis clique sur OK. Crée une autre variable "chasseur".

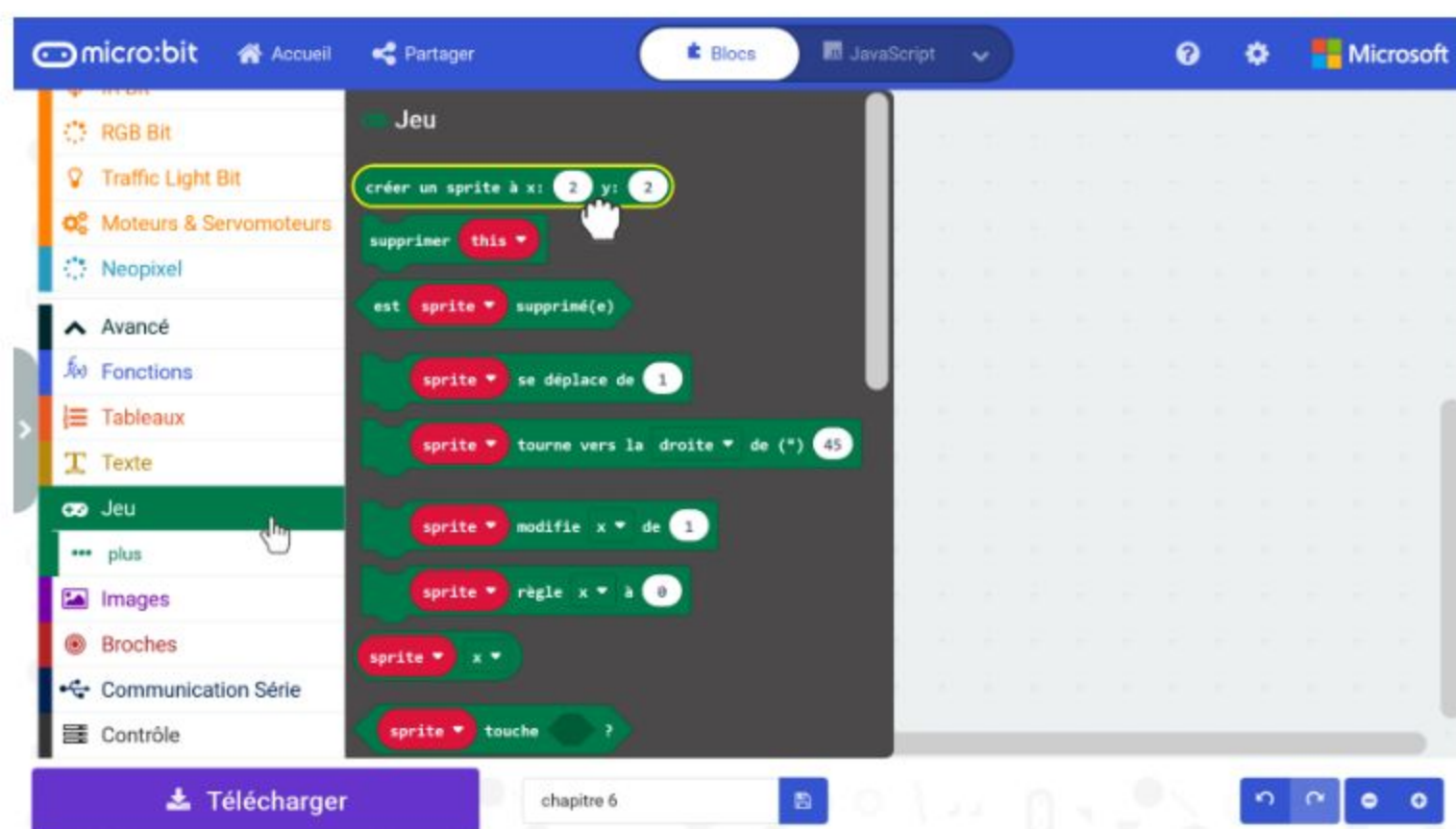


Étape 3 Clique sur la catégorie [**Variables**] et sélectionne le bloc [**définir _ à _**]. Duplique le bloc [**définir _ à _**] et imbrique-le dans le bloc [**au démarrage**]. Sélectionne les variables "chasseur" et "joueur" respectivement.

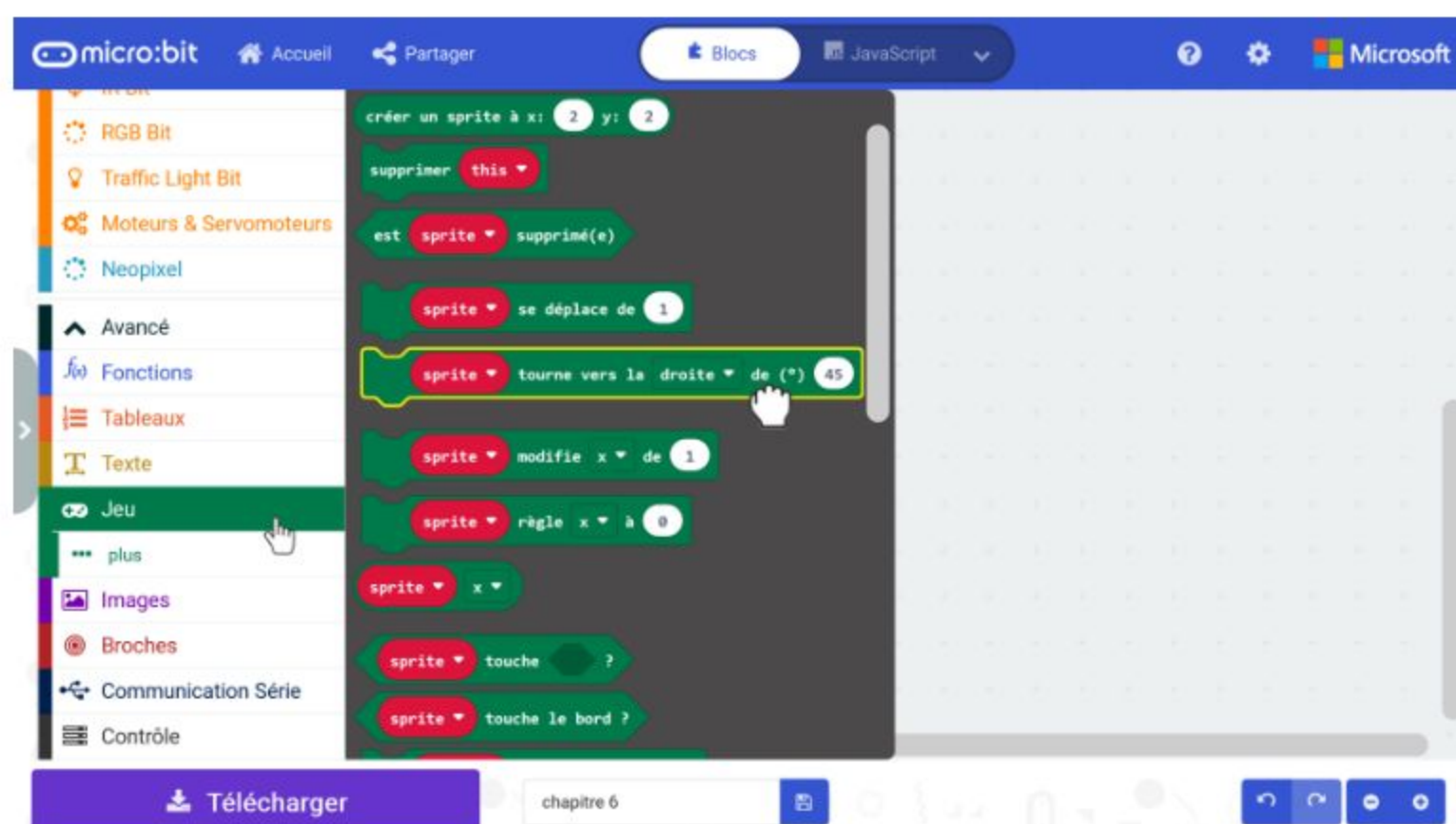


CHAPITRE 6 : C'est Toi le Chat !

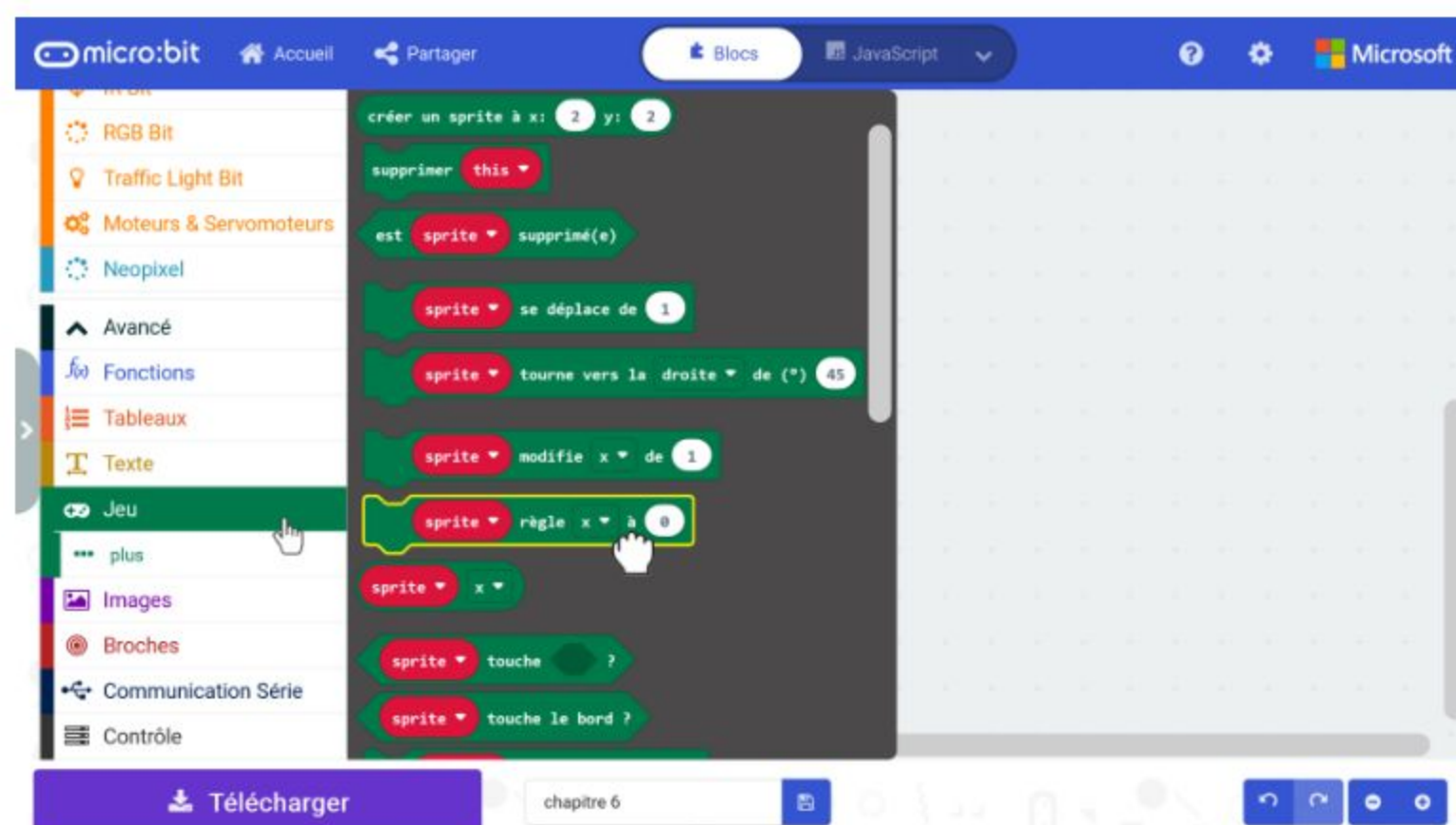
Étape 4 Clique sur [**Avancé**] puis sur la catégorie [**Jeu**]. Sélectionne le bloc [**créer un sprite à x: _ y: _**]. Duplique-le et insère-les dans les blocs [**définir _ à _**]. Saisis les valeurs **x:0 y:5** pour "chasseur" et **x:2 y:0** pour "joueur".



Étape 5 Clique sur la catégorie [**Avancé**] : [**Jeu**] et sélectionne le bloc [**_ tourne vers la _ de (°) _**]. Imbrique-le dans le bloc [**au démarrage**]. Sélectionne la variable "joueur" et saisis la valeur **90 (°)**.



Étape 6 Clique sur la catégorie [**Avancé**] : [**Jeu**] et sélectionne le bloc [**_ modifie _ de _**]. Sélectionne la variable "joueur", change "x" en "luminosité" et saisis la valeur **50**.

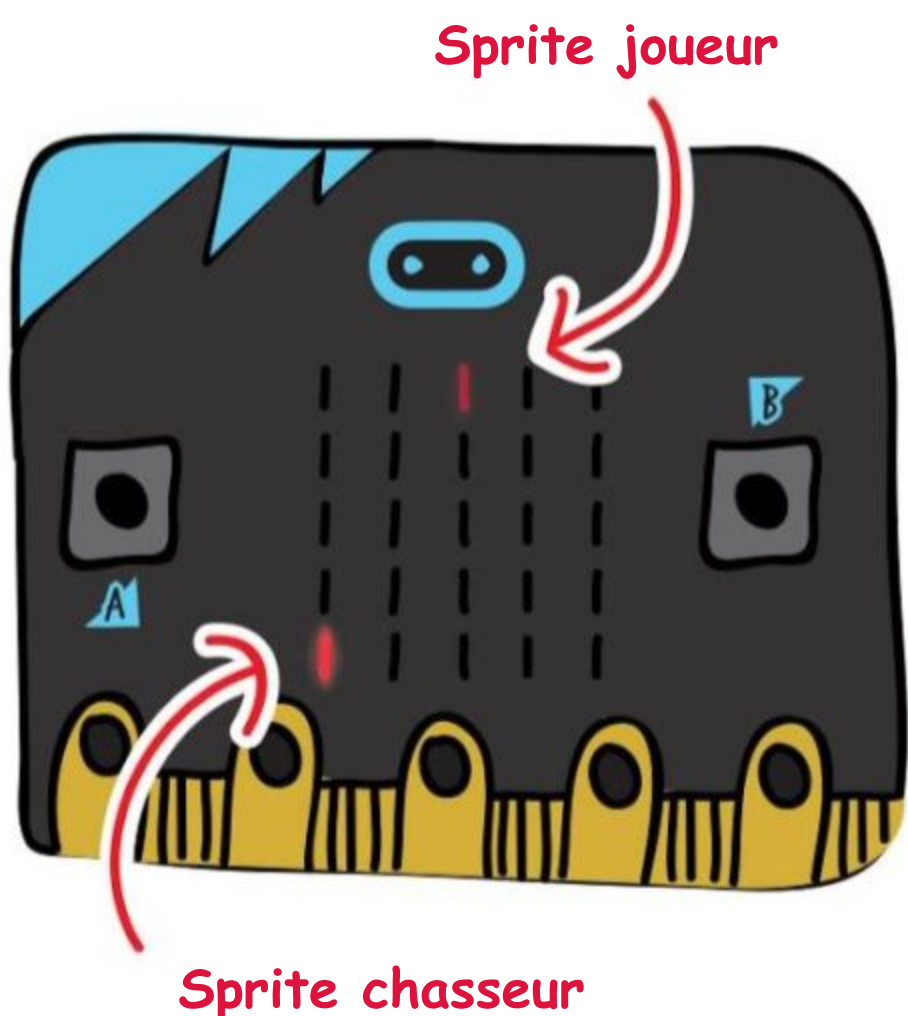




Étape 7 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**]. Duplique-le et change "A" en "B" pour le second bloc [**lorsque le bouton _ est pressé**].



Étape 8 Clique sur la catégorie [**Jeu**] et sélectionne le bloc [**_ se déplace de _**]. Duplique-le et imbrique-les dans les blocs [**lorsque le bouton _ est pressé**]. Sélectionne la variable "joueur" puis saisis la valeur -1 pour l'un (lorsque le bouton A est pressé) et 1 pour l'autre (lorsque le bouton B est pressé).



Flashe le code sur EDU:BIT. As-tu remarqué que lorsque tu appuies sur le bouton bleu (bouton B), la lumière se déplace vers le bas ? C'est le sprite joueur.

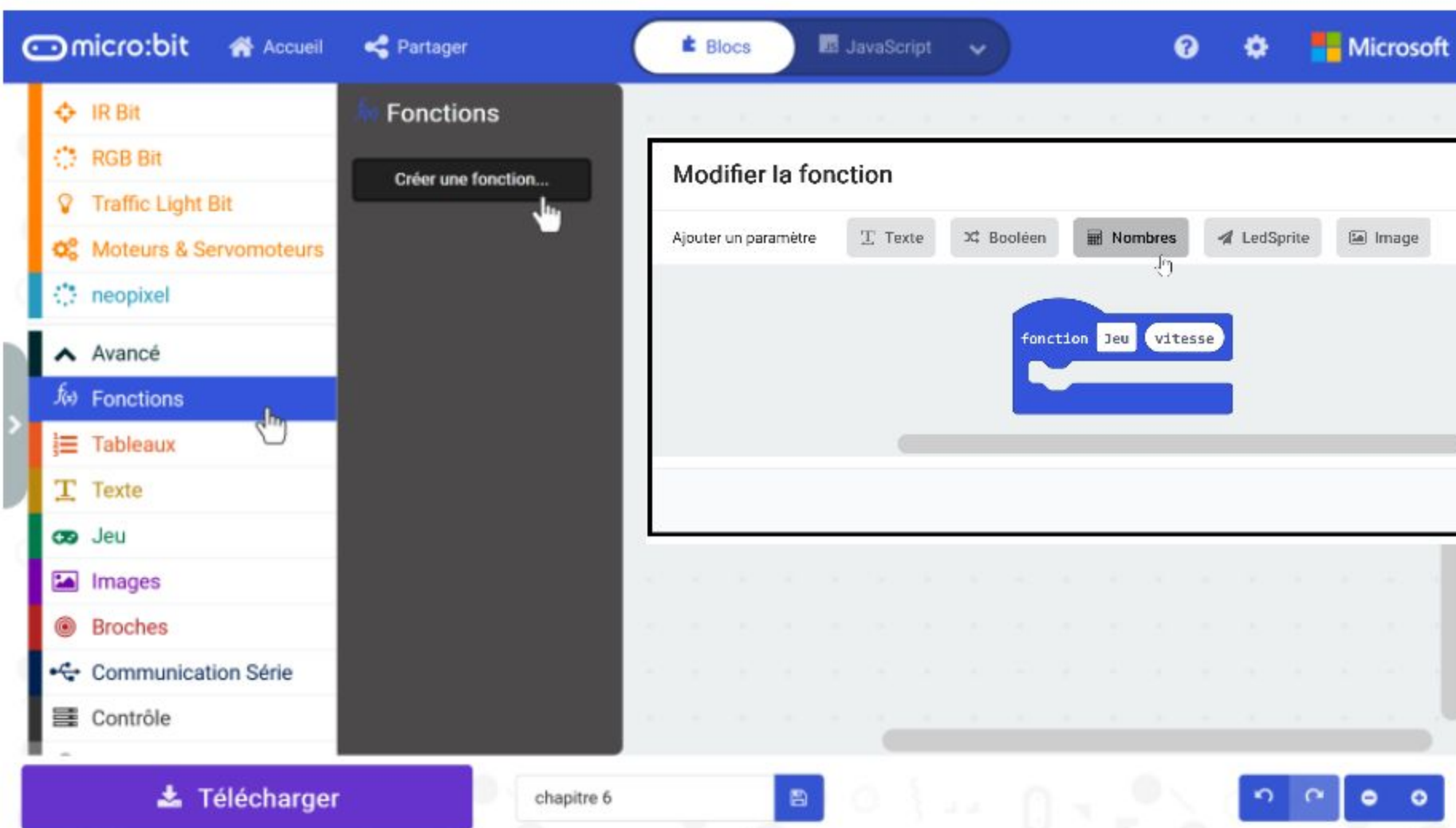
Un sprite est comme un "personnage LED" que tu peux contrôler.

Que se passe-t-il si tu appuies sur le bouton jaune (bouton A) ?



CHAPITRE 6 : C'est Toi le Chat !

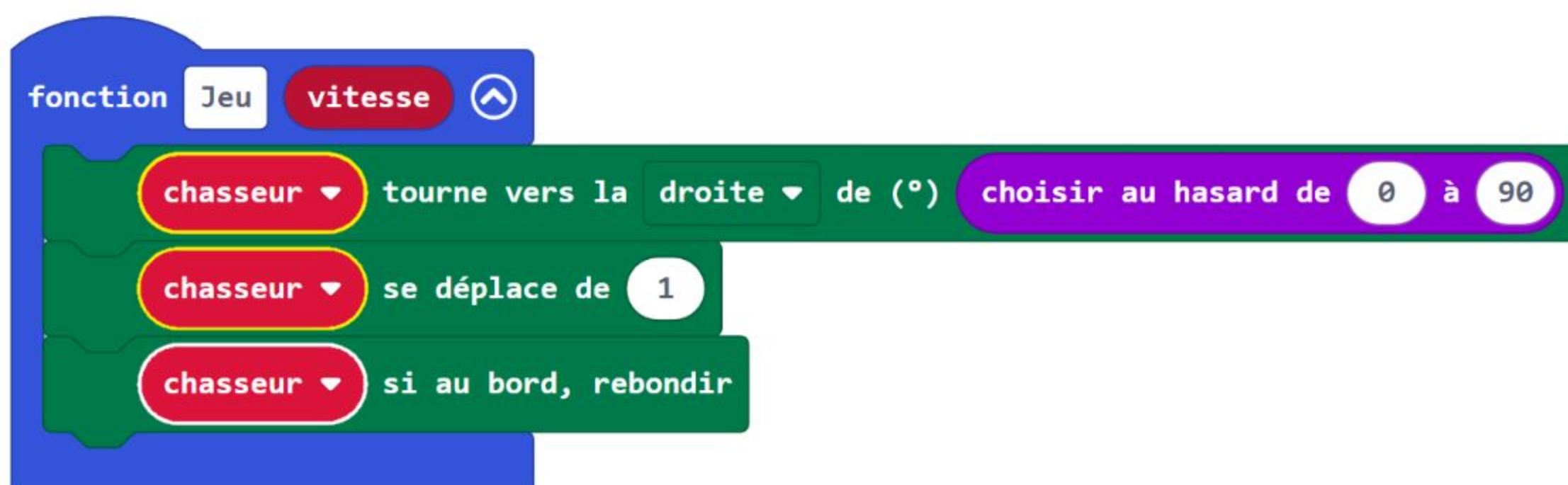
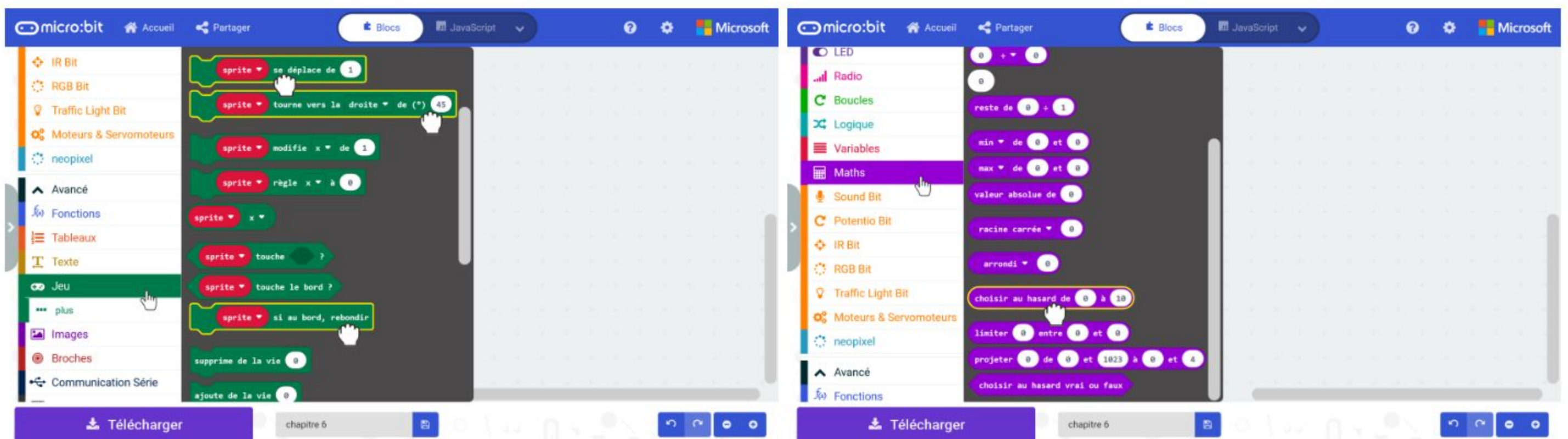
Étape 9 Clique successivement sur [**Avancé**], sur la catégorie [**Fonctions**], puis sur [**Créer une fonction...**]. Dans la fenêtre “Modifier la fonction”, renomme “doSomething” en “**Jeu**”. Ensuite clique sur [**Nombres**] pour ajouter un paramètre et renomme “num” en “**vitesse**” dans le bloc de la fonction. Enfin, clique sur “**Terminé**”.



Ce bloc fonction apparaît dans l'espace de travail.

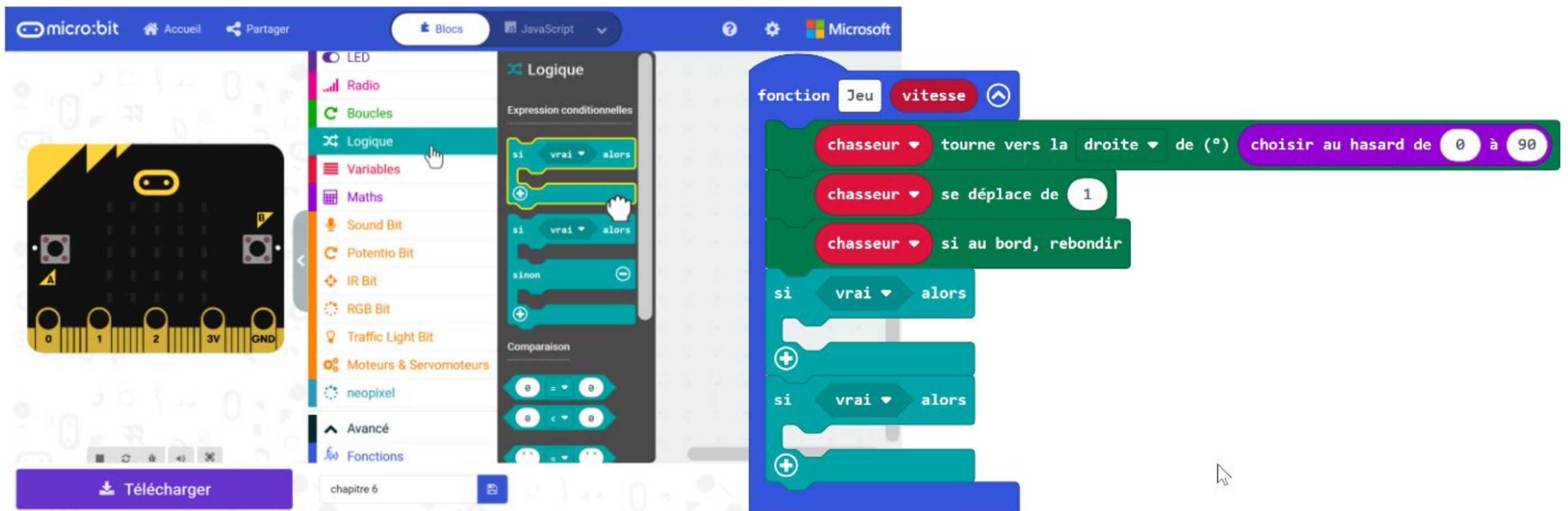


Étape 10 Continue à construire ton code en ajoutant des blocs des catégories [**Avancé**] : [**Jeu**] et [**Maths**] comme montré ci-dessous. Pense à changer la variable en “**chasseur**” et la valeur en **90**.

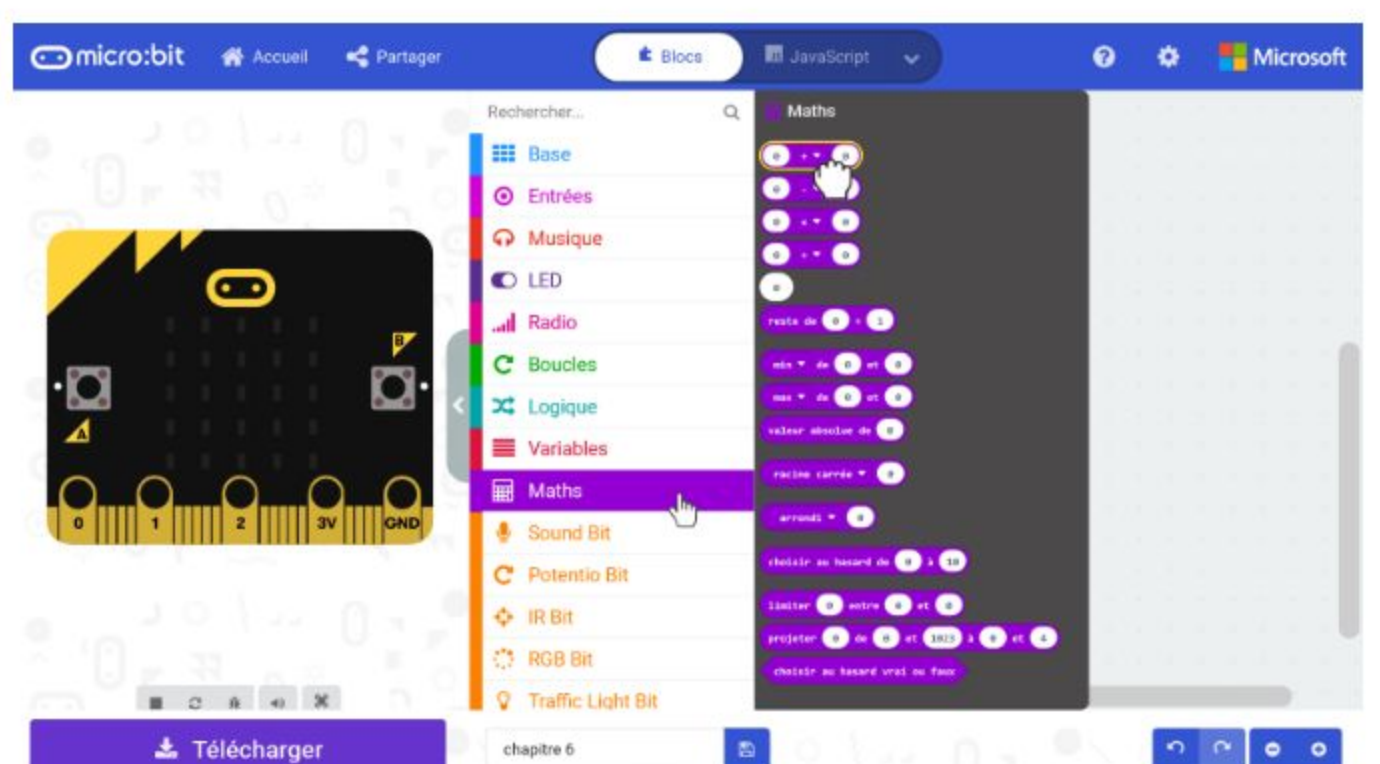
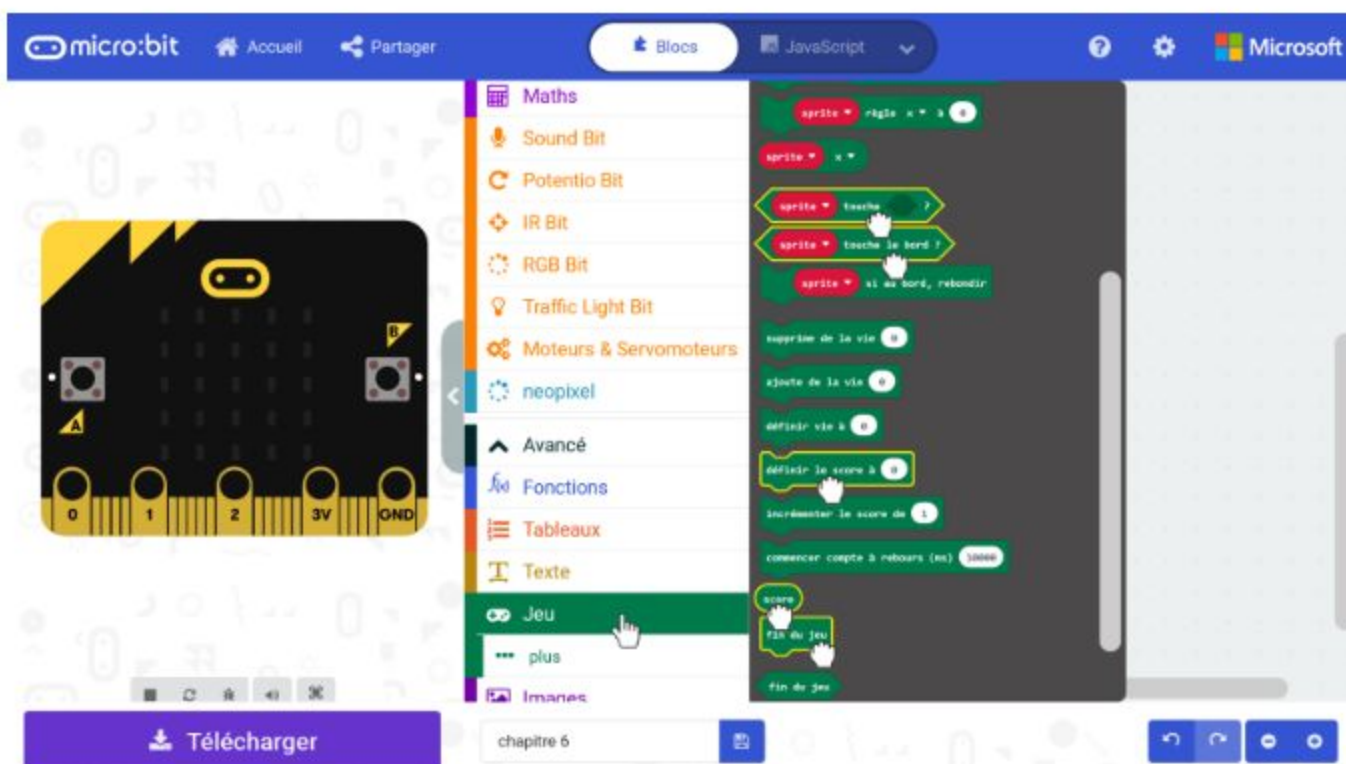




Étape 11 Ajoute deux blocs [si-alors] de la catégorie [Logique] à ton code.



Étape 12 Continue à construire ton code en ajoutant des blocs des catégories [Avancé] : [Jeu] et [Maths] comme montré ci-dessous.



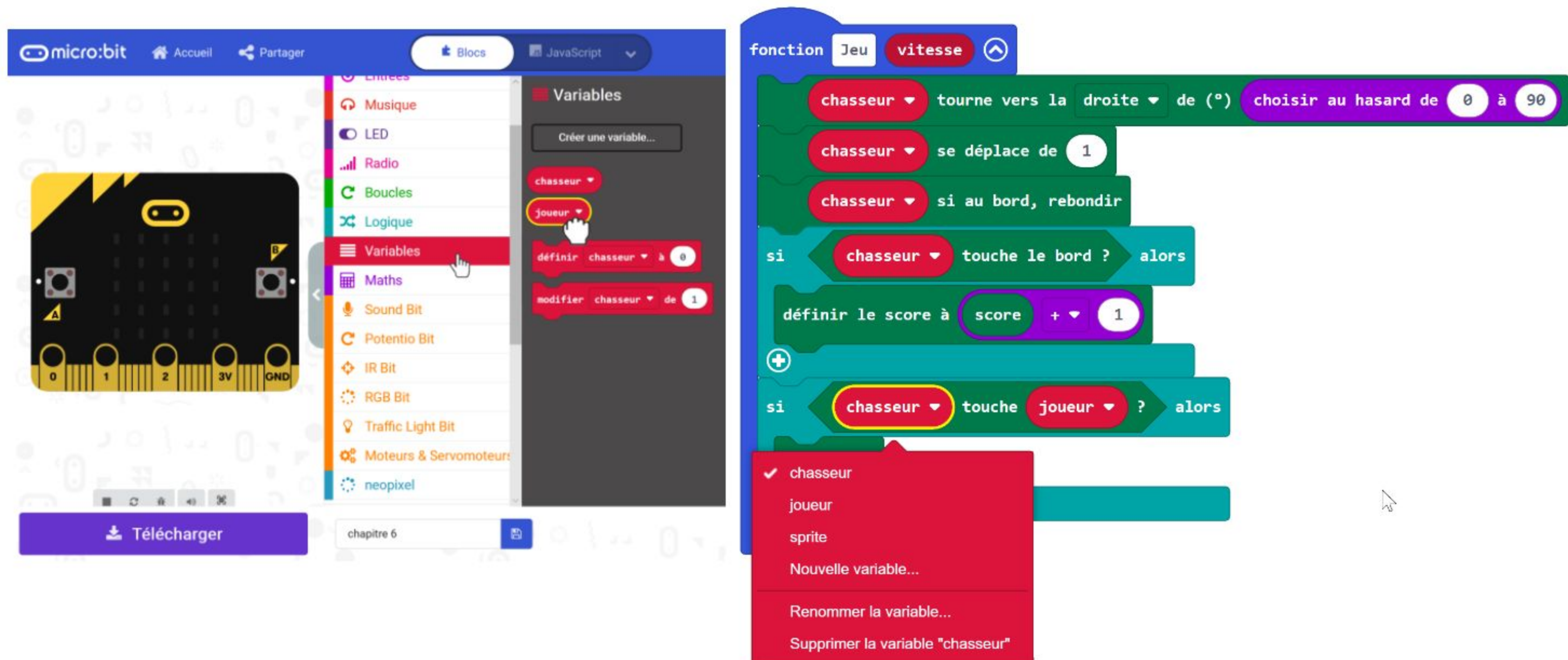
```

fonction Jeu vitesse
  chasseur tourne vers la droite de (°) choisir au hasard de 0 à 90
  chasseur se déplace de 1
  chasseur si au bord, rebondir
  si sprite touche le bord ? alors
    définir le score à score + 1
  si sprite touche ? alors
    fin du jeu
  
```

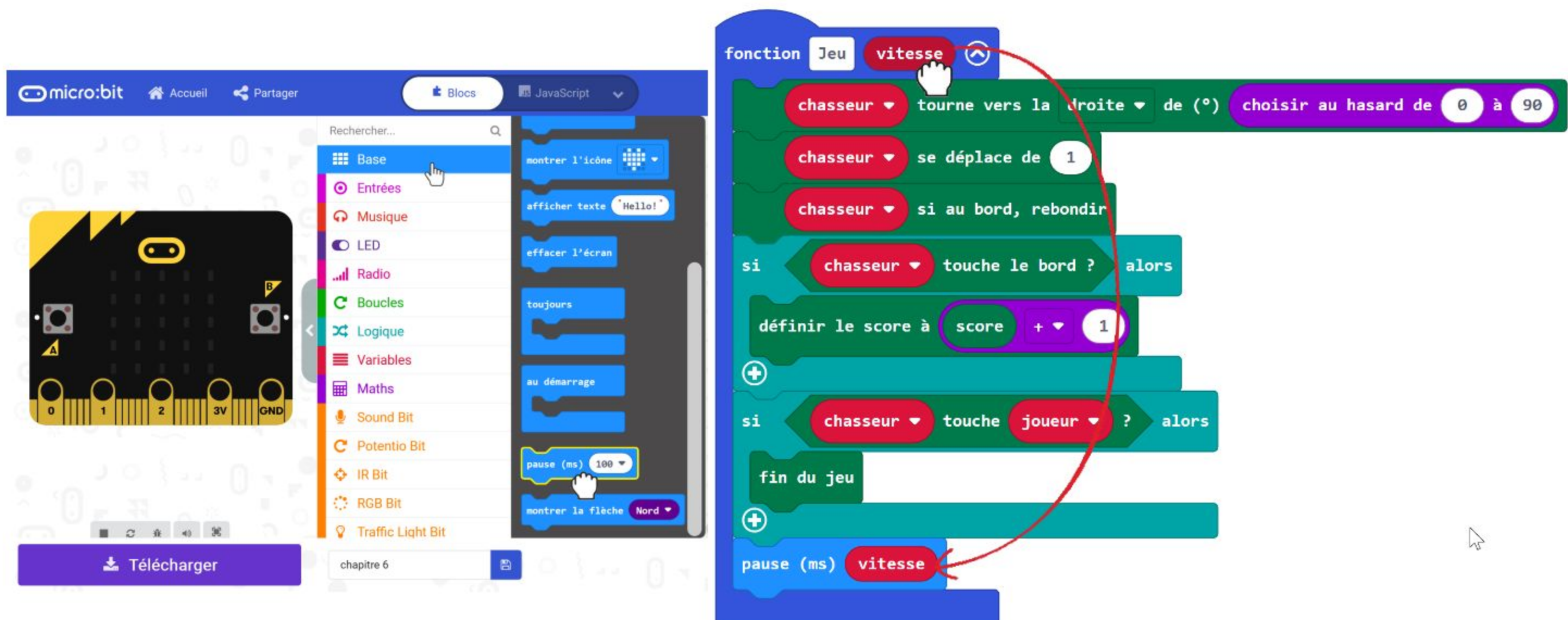
Garde le cap !

CHAPITRE 6 : C'est Toi le Chat !

Étape 13 Change le bloc **[sprite]** en **“chasseur”** en cliquant dessus et en sélectionnant **“chasseur”** dans la liste. Clique sur la catégorie **[Variables]** et sélectionne le bloc **[joueur]**. Insère-le dans le bloc **[_ touche _ ?]**.



Étape 14 Clique sur la catégorie **[Base]** et sélectionne le bloc **[pause _ (ms)]**. Ajoute-le au code. Clique sur le bloc **[vitesse]** dans l'en-tête du bloc fonction et maintiens le clic, fais le glisser et insère-le dans le bloc **[pause _ (ms)]**.



Tu peux rendre le jeu plus excitant en ajoutant une mélodie lorsque le chasseur touche le joueur. Peux-tu deviner quels blocs ajouter et où les placer ?



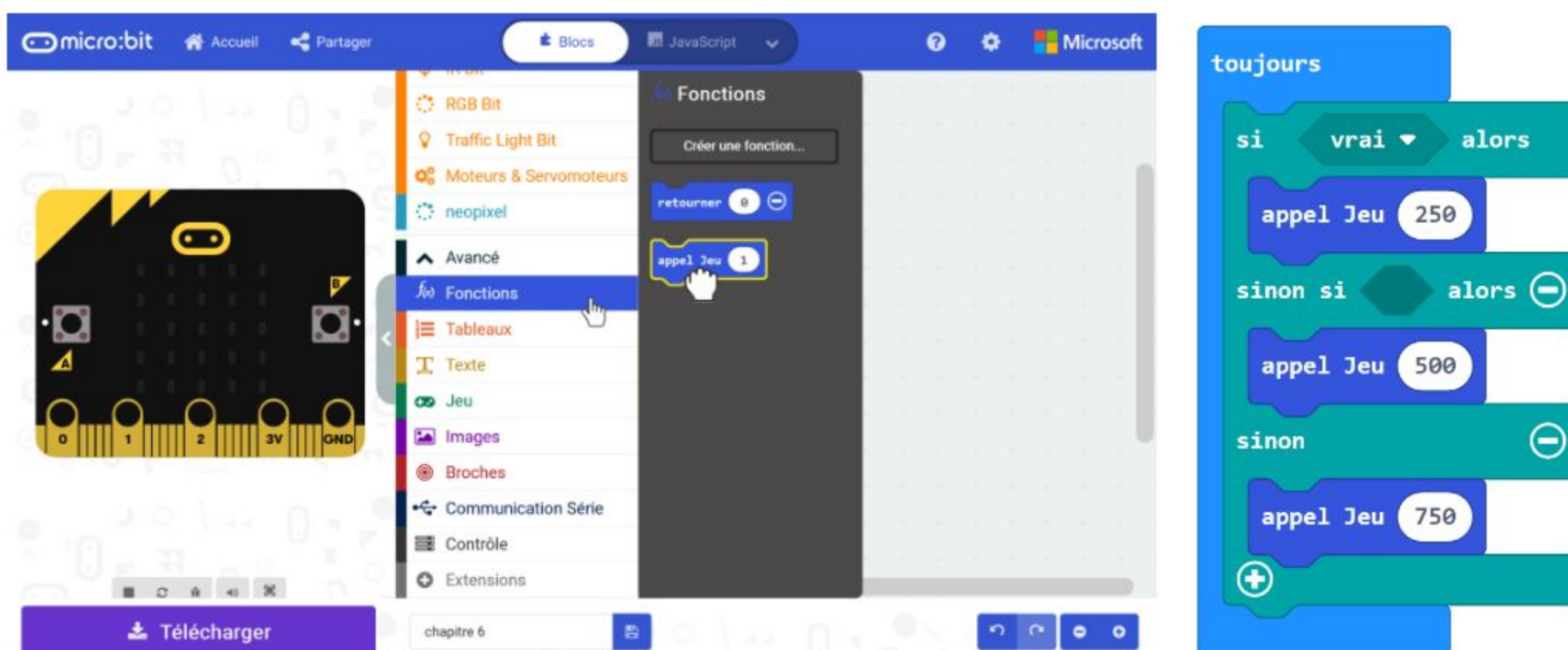


Maintenant, ajoutons différents niveaux de difficulté au jeu !

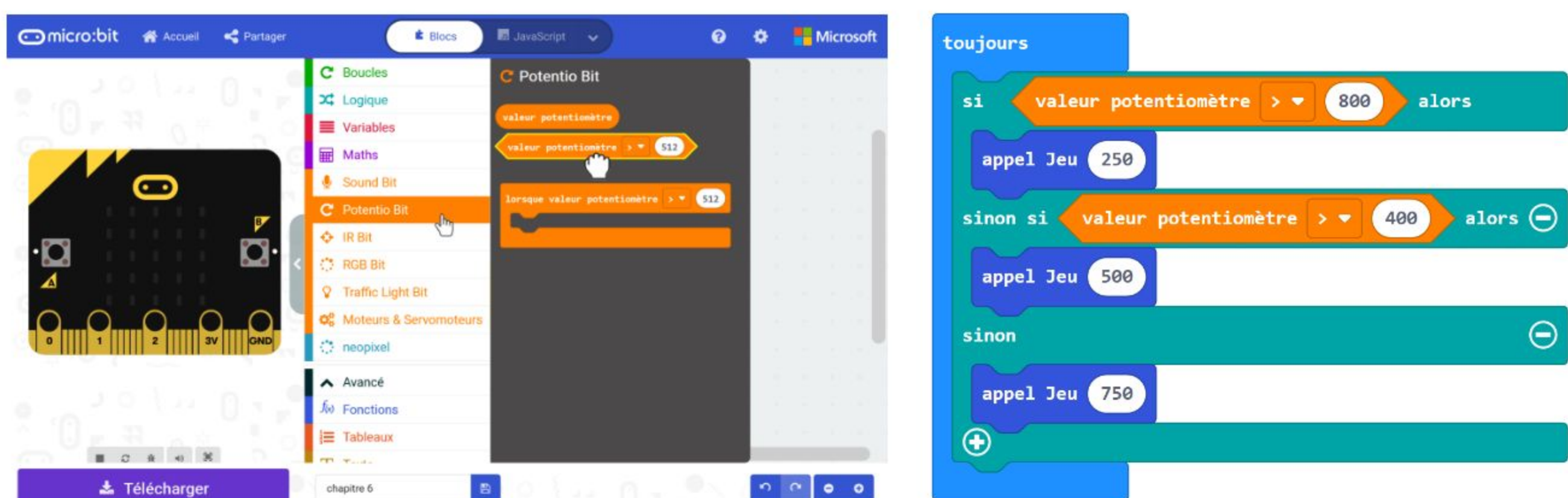
Étape 15 Clique sur la catégorie [**Logique**] et sélectionne le bloc [**si-alors-sinon**]. Imbrique-le dans le bloc [**toujours**]. Clique sur le bouton + pour ajouter un test.



Étape 16 Clique sur la catégorie [**Fonctions**] et sélectionne le bloc [**appel Jeu _**]. Duplique-le deux fois et imbrique-les dans les emplacements du bloc [**si-alors-sinon**]. Change la valeur de chaque bloc [**appel Jeu _**] en **250**, **500** et **750** respectivement.



Étape 17 Clique sur la catégorie [**Potentio Bit**] et sélectionne le bloc [**valeur potentiomètre > _**]. Duplique-le et insère-les dans les conditions du bloc [**si-alors-sinon**]. Change la valeur en **800** pour le premier bloc et **400** pour le second.



CHAPITRE 6 : C'est Toi le Chat !

Voici le code complet :

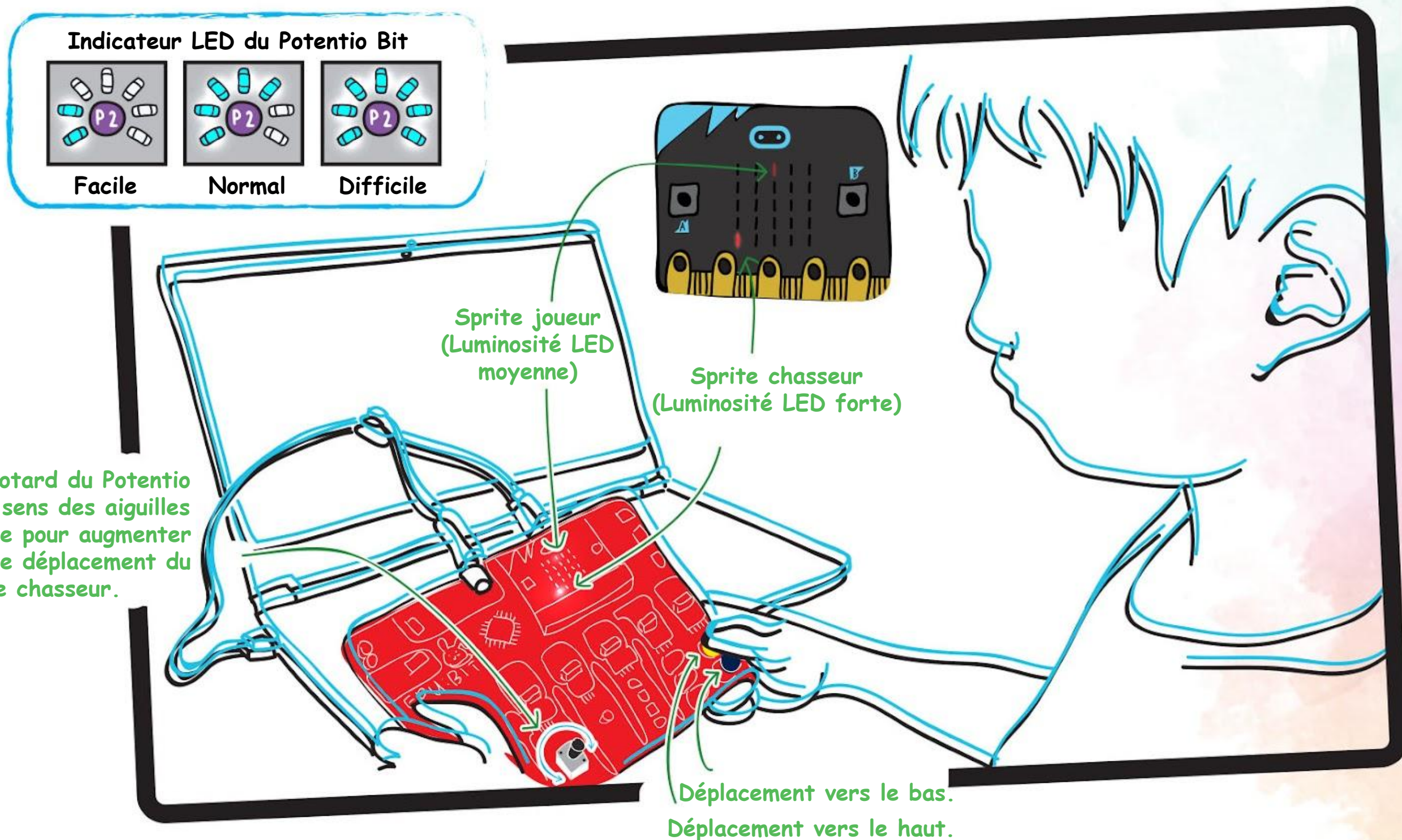
The image displays a Scratch script for a game called 'C'est Toi le Chat!'. The script is organized into several sections, each with a corresponding explanatory text box on the left. The code is as follows:

- au démarrage** (when green flag clicked):
 - commencer compte à rebours (ms) 30000
 - définir chasseur à créer un sprite à x: 0 y: 5
 - définir joueur à créer un sprite à x: 2 y: 0
 - joueur tourne vers la droite de (°) 90
 - joueur modifie luminosité de 50
- lorsque le bouton A est pressé** (when button A is pressed):
 - joueur se déplace de -1
- lorsque le bouton B est pressé** (when button B is pressed):
 - joueur se déplace de 1
- fonction Jeu vitesse** (function):
 - chasseur tourne vers la droite de (°) choisir au hasard de 0 à 90
 - chasseur se déplace de 1
 - chasseur si au bord, rebondir
 - si chasseur touche le bord ? alors
 - définir le score à score + 1
 - si chasseur touche joueur ? alors
 - démarrer la mélodie wawawawaa répétition une fois
 - fin du jeu
 - pause (ms) vitesse
- si valeur potentiomètre > 800 alors** (when potentiometer value > 800):
 - appel Jeu 250
- sinon si valeur potentiomètre > 400 alors** (otherwise if potentiometer value > 400):
 - appel Jeu 500
- sinon** (otherwise):
 - appel Jeu 750

Étape 18 Flashe le code sur EDU:BIT et amuse-toi à jouer à *C'est Toi le Chat!* avec tes amis.

Jouons

C'est Toi le Chat !



COMMENT JOUER :

- Au lancement, le sprite chasseur se déplace dans des directions aléatoires.
- Déplacer le sprite joueur vers le haut ou vers le bas pour éviter le sprite chasseur. Appuyer sur le bouton jaune (bouton A) pour le déplacer vers le haut et le bouton bleu (bouton B) pour le déplacer vers le bas.
- Le jeu s'arrête si le sprite chasseur touche le sprite joueur, ou après 30 secondes de jeu, durée maximale d'une partie.
- Chaque fois que le sprite chasseur touche le bord, le score augmente de 1 point. Le joueur avec le plus grand score est le vainqueur ! Amuse-toi~

Astuces !

*#1 Pour obtenir un score plus élevé dans la limite du temps imparti, tu peux augmenter la vitesse du chasseur, qui touchera alors plus souvent le bord.
#2 Lorsque la partie est terminée, tu peux appuyer simultanément sur les boutons A+B pour commencer une nouvelle partie. C'est une fonction intégrée des blocs de la catégorie [Jeu].*

DÉCHIFFRE

LE CODE

En programmation, on utilise des **instructions conditionnelles Si** pour définir des prises de décision. En MakeCode, on utilise les blocs [**si-alors**] ou [**si-alors-sinon**] de la catégorie [**Logique**] pour définir les conditions. Le programme teste alors la condition, et si elle est VRAIE, il exécute le code imbriqué dans la partie “si” du bloc. Sinon, le test est FAUX, et ce code n’est pas exécuté, le programme passe à la suite.

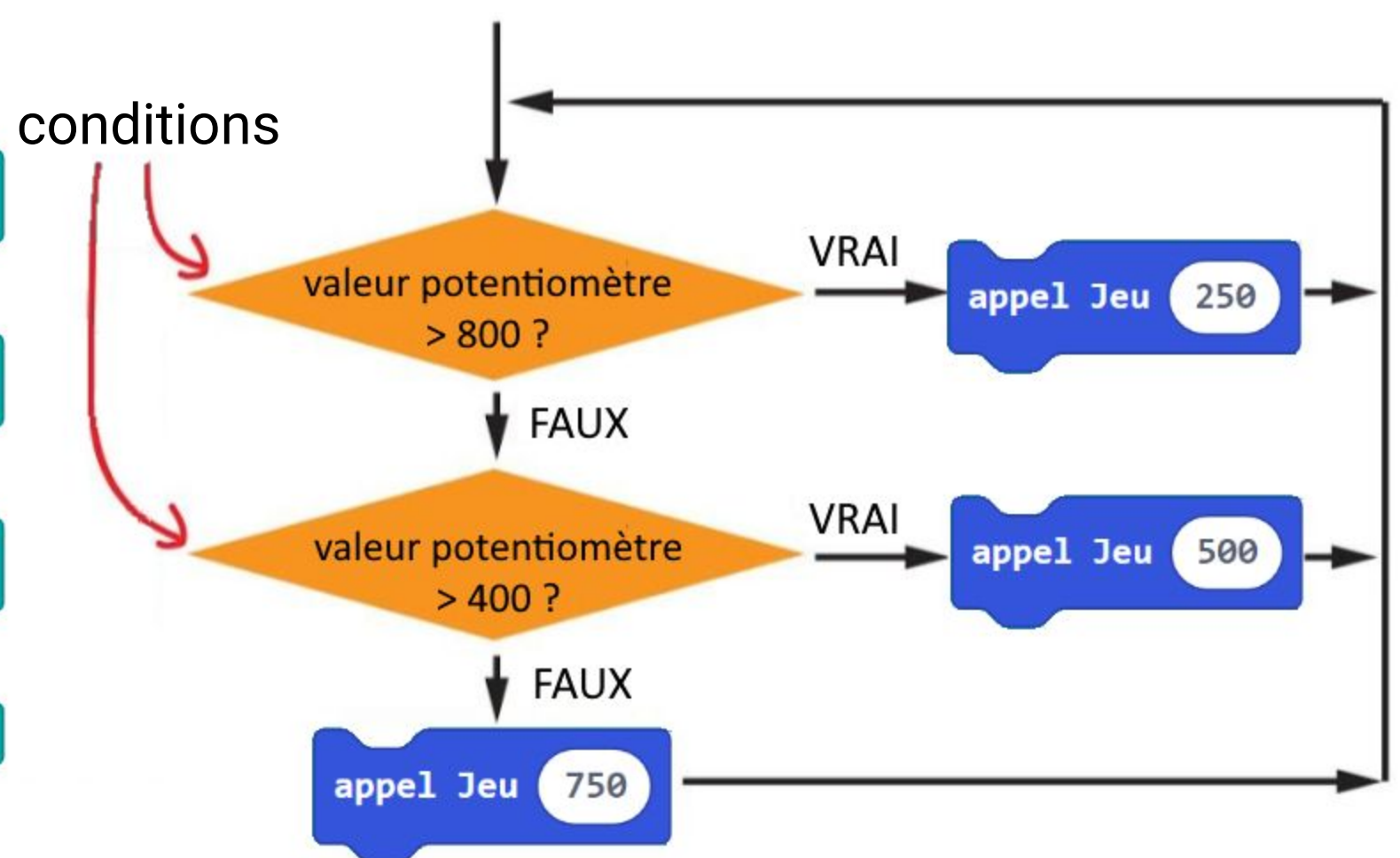
```
si chasseur touche joueur ? alors
  démarrer la mélodie wawawawaa répétition une fois
  fin du jeu
```

Si la condition est vraie (ie : le sprite chasseur touche le sprite joueur), **alors** faire ceci : (jouer la mélodie wawawawaaa, puis afficher l’animation “Fin de jeu”).

En cas de conditions multiples, le programme évaluera les conditions successivement, du haut vers le bas, puis exécute le code correspondant à la première condition VRAIE. Donc, si plusieurs conditions sont vraies, la priorité va à la plus haute dans le code, la première à être testée.

Par exemple, ce code extrait du jeu détermine la vitesse de mouvement du sprite chasseur en comparant la valeur du potentiomètre à des valeurs fixées.

```
toujours
  si valeur potentiometre > 800 alors
    appel Jeu 250
  sinon si valeur potentiometre > 400 alors
    appel Jeu 500
  sinon
    appel Jeu 750
```



Si valeur potentiomètre > 800, appeler la fonction Jeu (avec la variable vitesse définie à 250 ms),
Sinon si valeur potentiomètre > 400, appeler la fonction Jeu (avec la variable vitesse définie à 500 ms),
Sinon appeler la fonction Jeu (avec la variable vitesse définie à 750 ms).

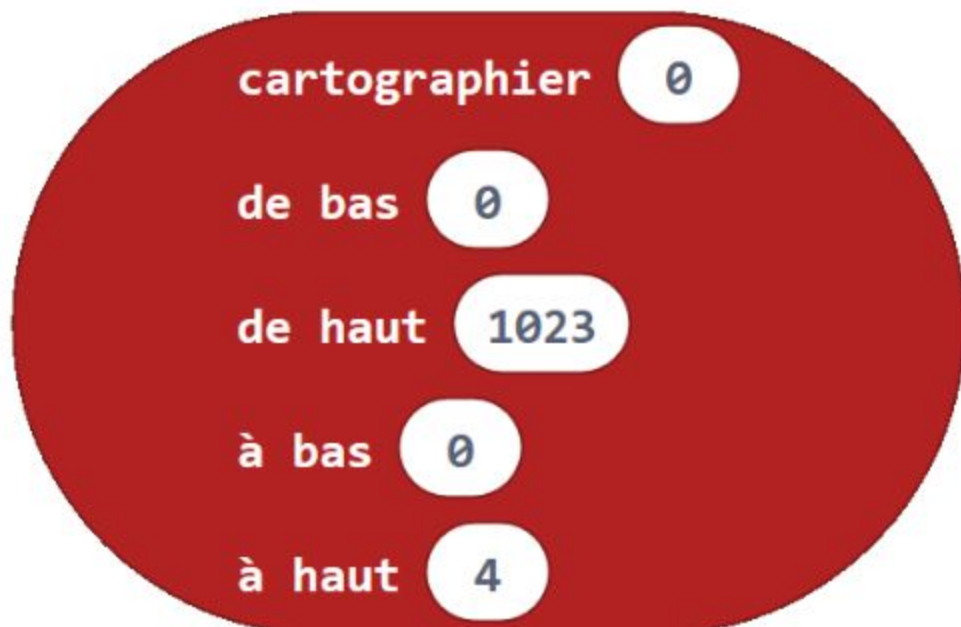


EXPLORE D'AUTRES BLOCS

#1 Utilise un bloc [Base] : [montrer nombre] avec un bloc [Potentio Bit] : [valeur potentiomètre] pour lire et afficher la valeur actuelle du potentiomètre.

montrer nombre valeur potentiomètre

#2 Le potentiomètre retourne une valeur comprise entre 0 et 1023. Tu peux utiliser le mappage à l'aide du bloc [cartographeur _ de bas _ de haut _ à bas _ à haut _] de la catégorie [Avancé] : [Broches] pour obtenir des valeurs variant dans un ensemble de nombres plus restreint.

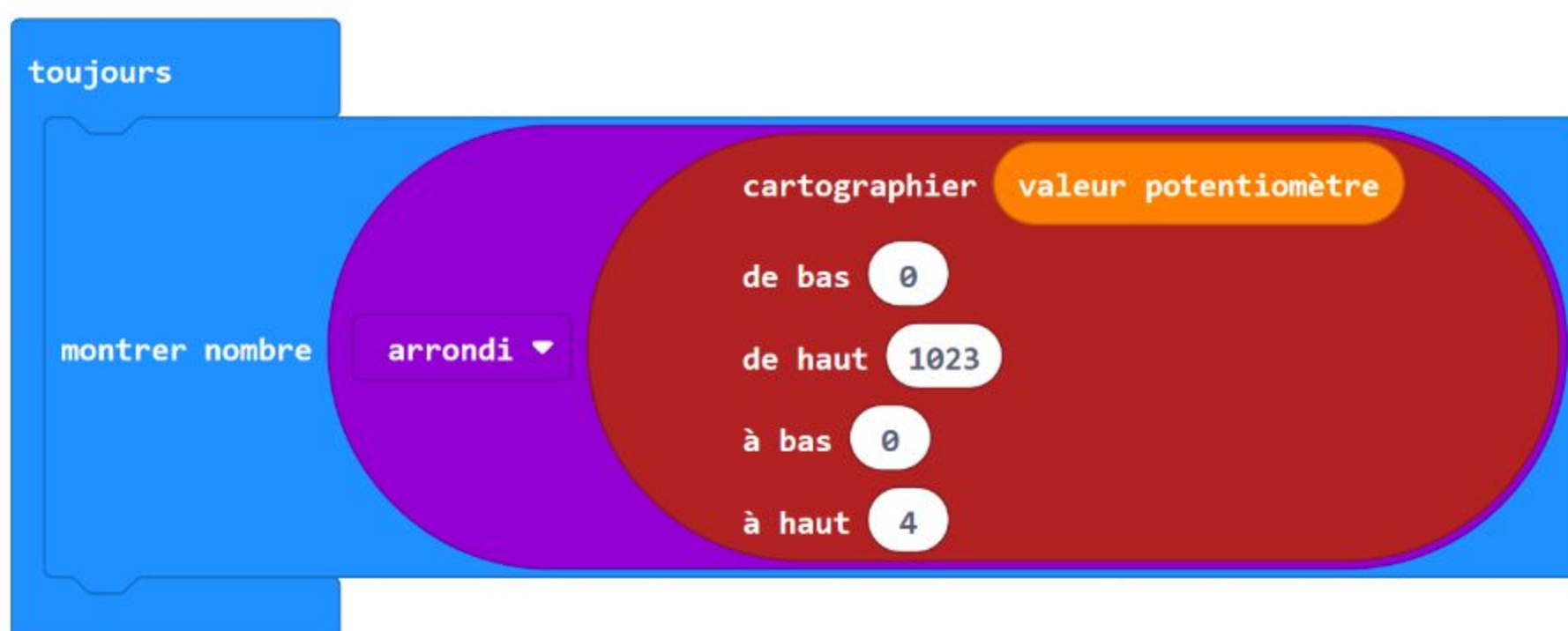


#3 Le bloc de mappage retournera un nombre décimal (par exemple : 1,68 ou 3,998). Pour obtenir une valeur approchée entière, on peut utiliser le bloc [arrondi _] de la catégorie [Maths].

arrondi ▾

0

Voici un exemple de code de mappage de l'ensemble des valeurs retournées par le potentiomètre (de 0 à 1023) à un ensemble d'entiers compris entre 0 et 7. La valeur sera affichée sur la matrice de LEDs.



N'oublie pas d'allumer EDU:BIT pour obtenir une valeur.



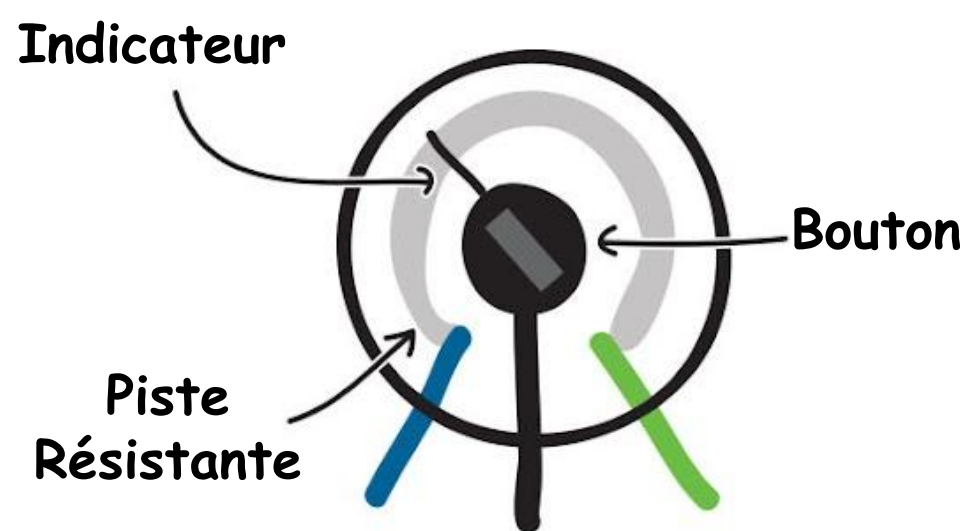
FAIT AMUSANT!



Les **potentiomètres**, parfois aussi appelés potards, sont des résistances variables dont la valeur de résistance peut être facilement réglée à l'aide de boutons par exemple.



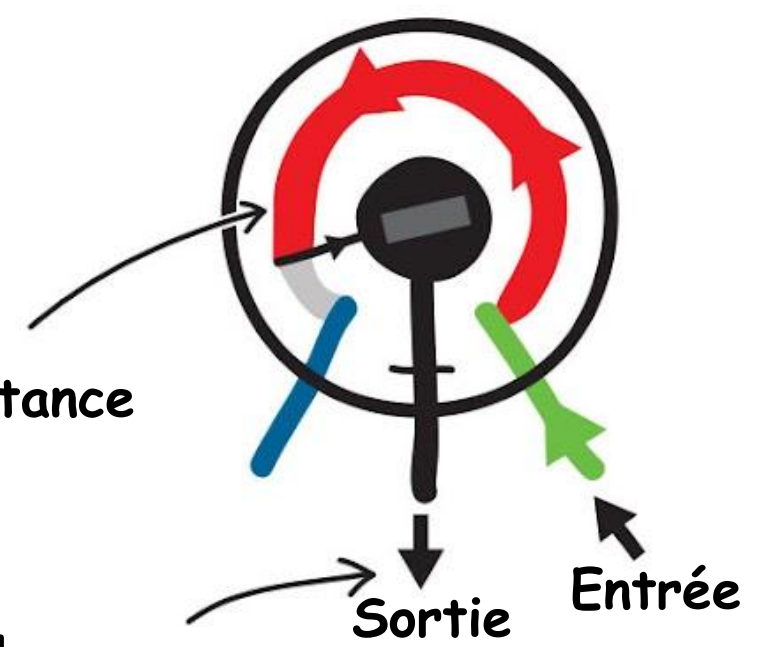
Si tu as un potentiomètre de 10 000 Ω (Ohms), tu peux avoir une résistance comprise entre 0 Ω et 10 000 Ω en tournant le bouton.



Résistance Faible



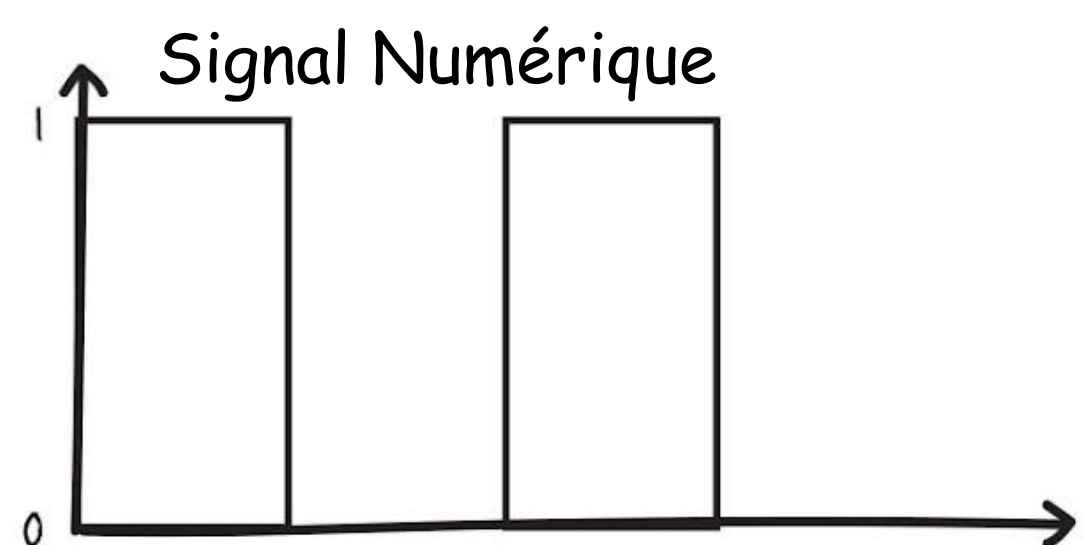
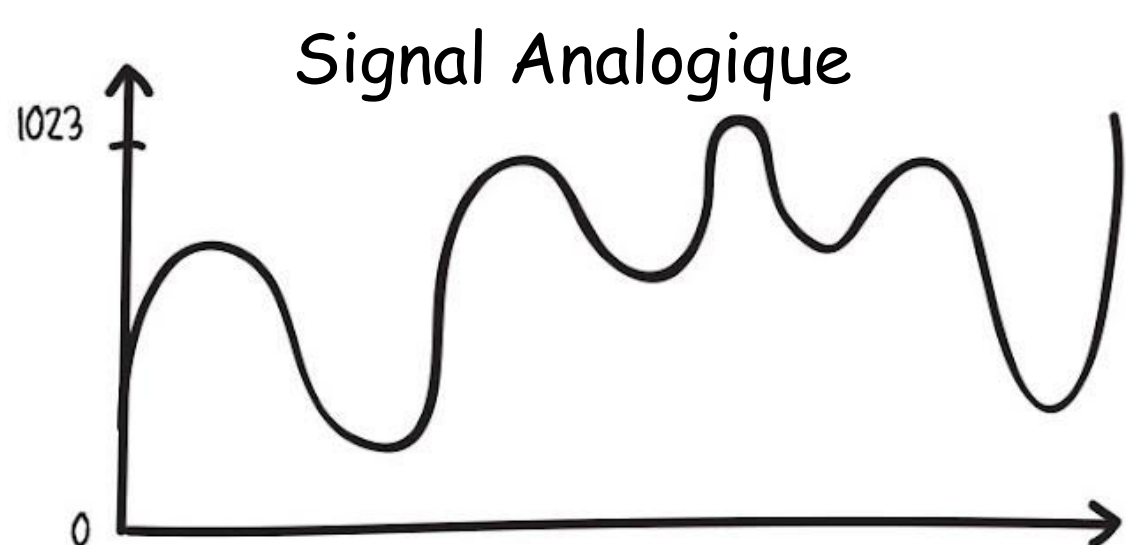
Résistance Forte



Voici des exemples d'utilisation :

- Régler le volume audio d'un haut-parleur
- Changer la fréquence d'une radio
- Régler la température d'un chauffe-eau

Le potentiomètre d'EDU:BIT est un dispositif d'entrée analogique. Il mesure le potentiel électrique et convertit la tension mesurée (entre 0 V et 3,3 V) en un entier compris entre 0 et 1023.



APPLICATION DÉFI

Programme EDU:BIT pour qu'elle serve de minuteur. Utilise le Potentio Bit pour définir la durée (entre 0 et 60 secondes), le bouton A pour l'activer et le bouton B pour le réinitialiser.	
au démarrage	Définir mode à 0.
lorsque le bouton A est pressé (jaune)	Définir mode à 1. Définir début à temps d'exécution (ms). Afficher l'icône sourire.
lorsque que le bouton B est pressé (bleu)	Définir mode à 0.
toujours	Vérifier mode SI mode = 0, alors définir durée à l'arrondi de la valeur du potentiomètre mappée dans un ensemble de nombres allant de 0 à 60, et afficher durée sur la matrice de LEDs. SINON SI mode = 1, vérifier si (temps d'exécution - début) > (durée x 1 000). Si c'est VRAI, jouer la mélodie wawawaaa et définir mode à 2. SINON afficher l'icône triste.

Voici quelques conseils pour toi :

#1 Tu dois créer trois variables : mode, début et durée.

#2 Le bloc temps d'exécution (ms) vient de la catégorie [Entrées].

#3 Utilise un bloc conditionnel pour déterminer si le temps est écoulé.



temps d'exécution (ms)

- ▼

début ▼

≥ ▼

durée ▼

x ▼

1000

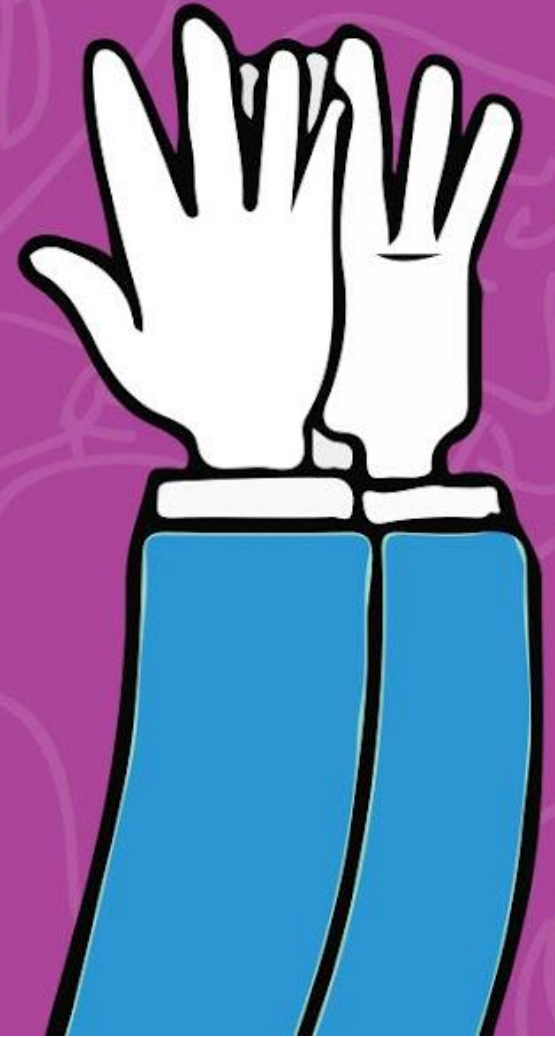
CHAPITRE 7

Écoutons les Applaudissements !

Sound Bit



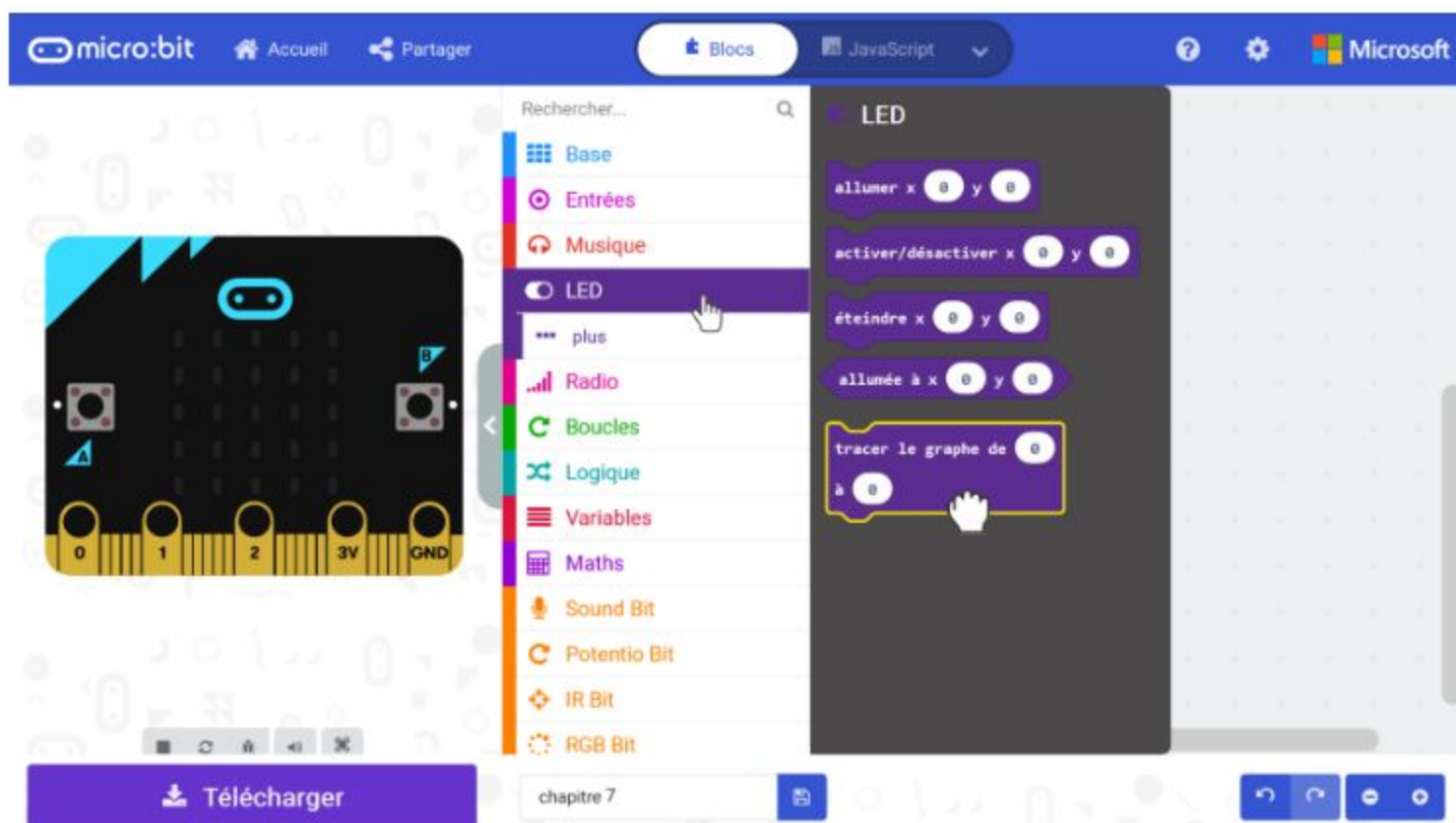
Scanne-moi !



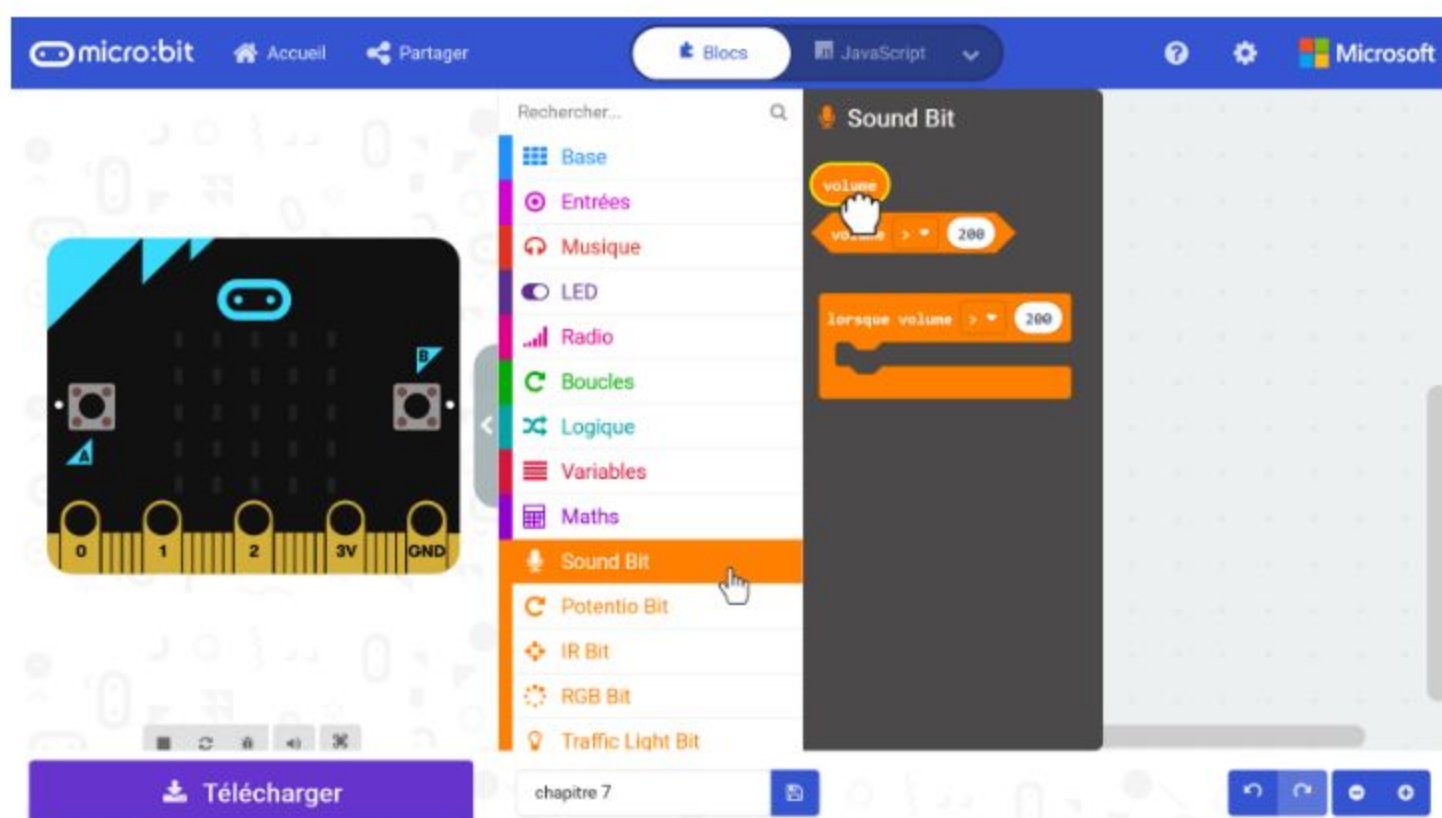


CODONS !

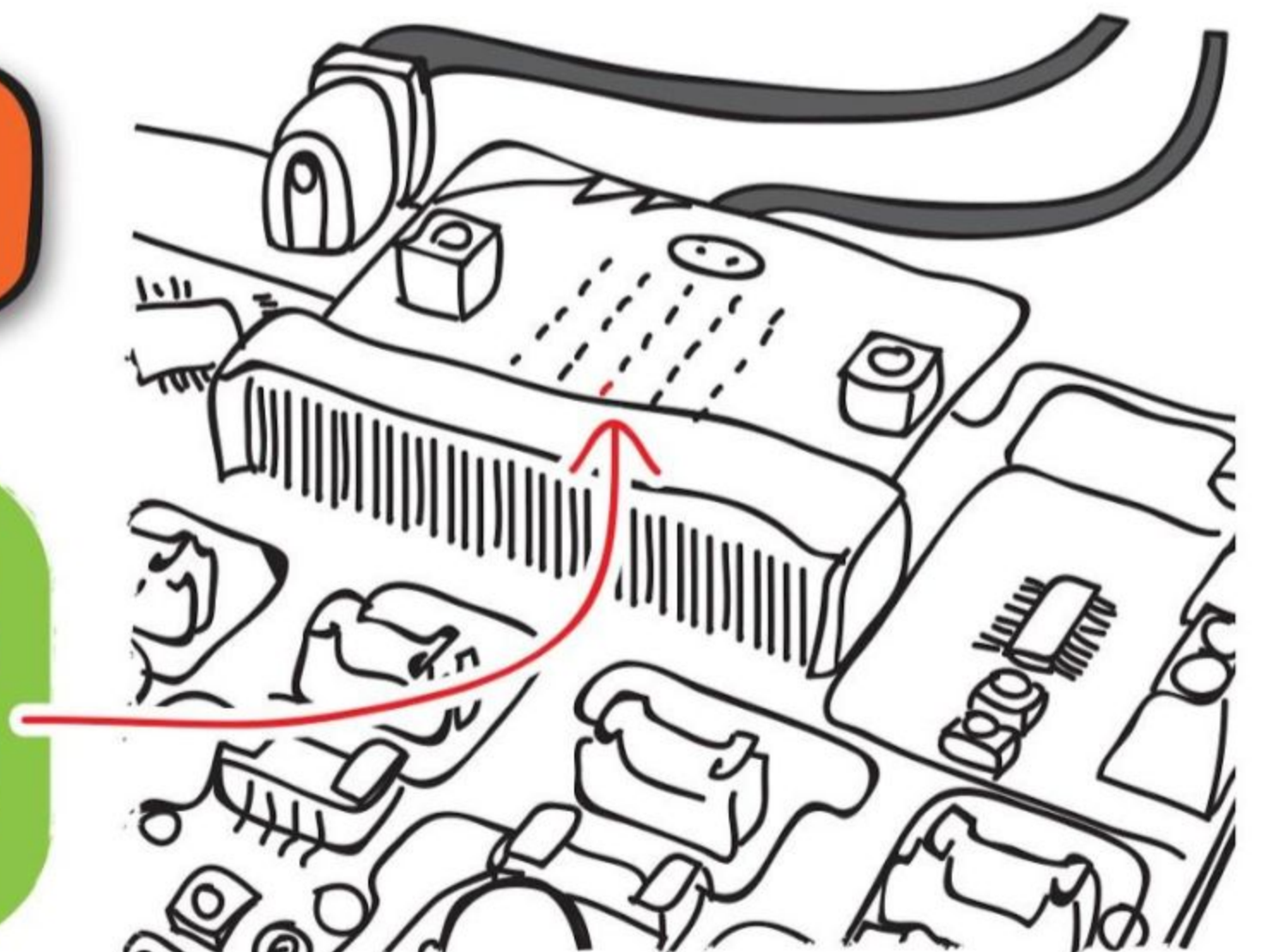
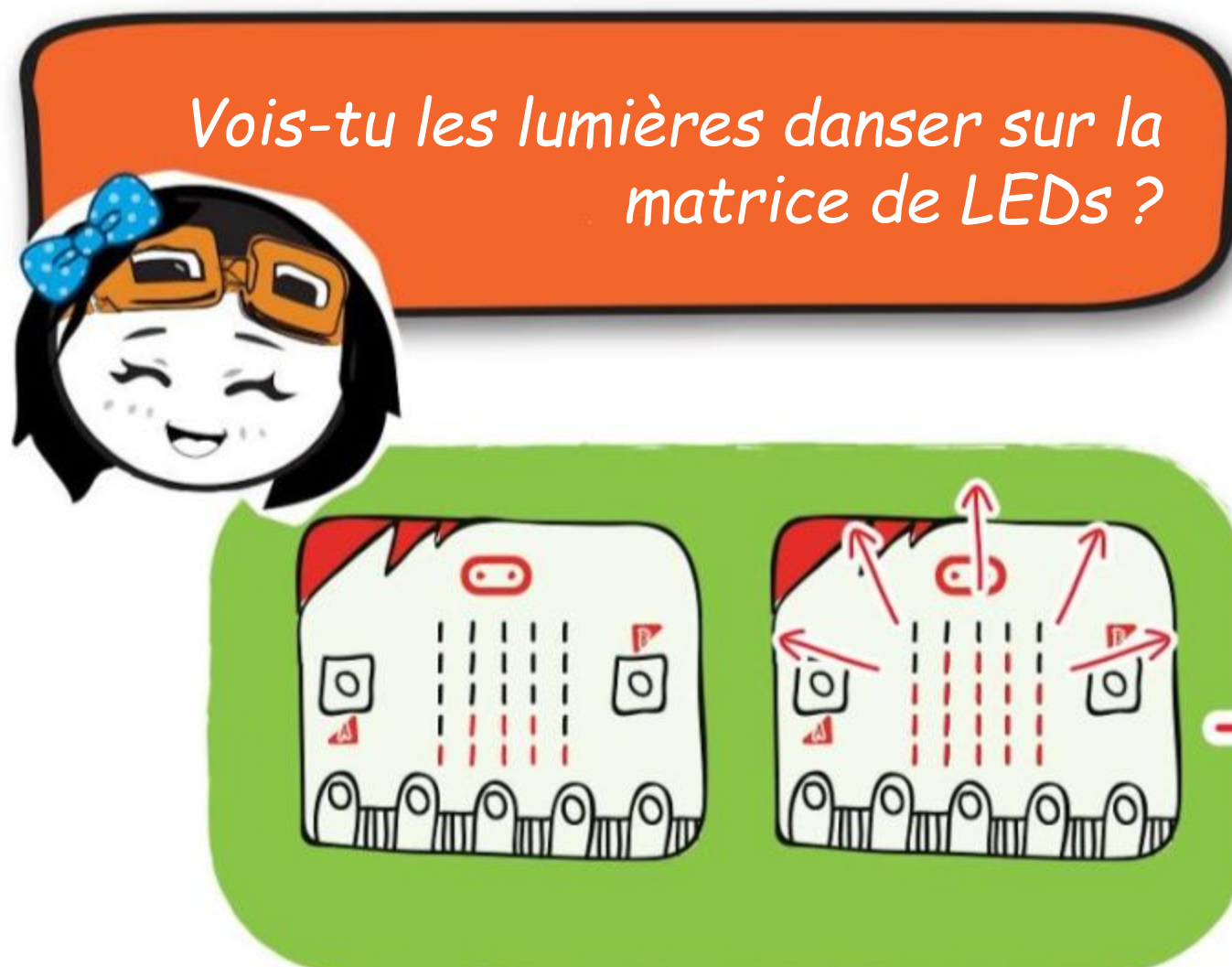
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT (voir page 40). Clique sur la catégorie [LED] et sélectionne le bloc [tracer le graphe de _ à _]. Imbrique-le dans le bloc [toujours].



Étape 2 Clique sur la catégorie [Sound Bit] et sélectionne le bloc [volume]. Insère-le dans le bloc [tracer le graphe de _ à _] et change la deuxième valeur en 1 023.



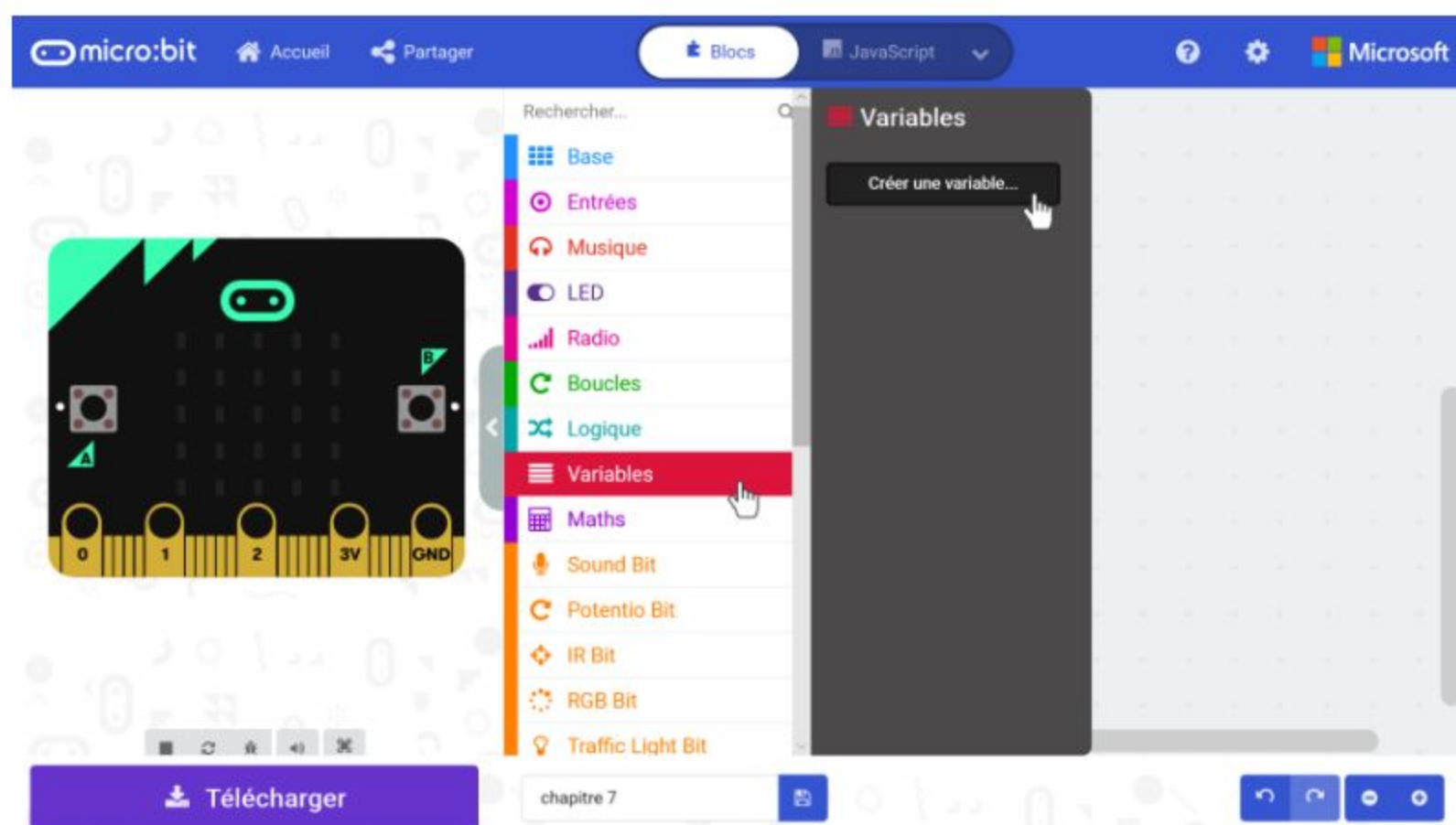
Étape 3 Flashe le code sur EDU:BIT. Observe la matrice de LEDs lorsque tu tapes dans tes mains ou tapes le rythme avec tes doigts sur la table.



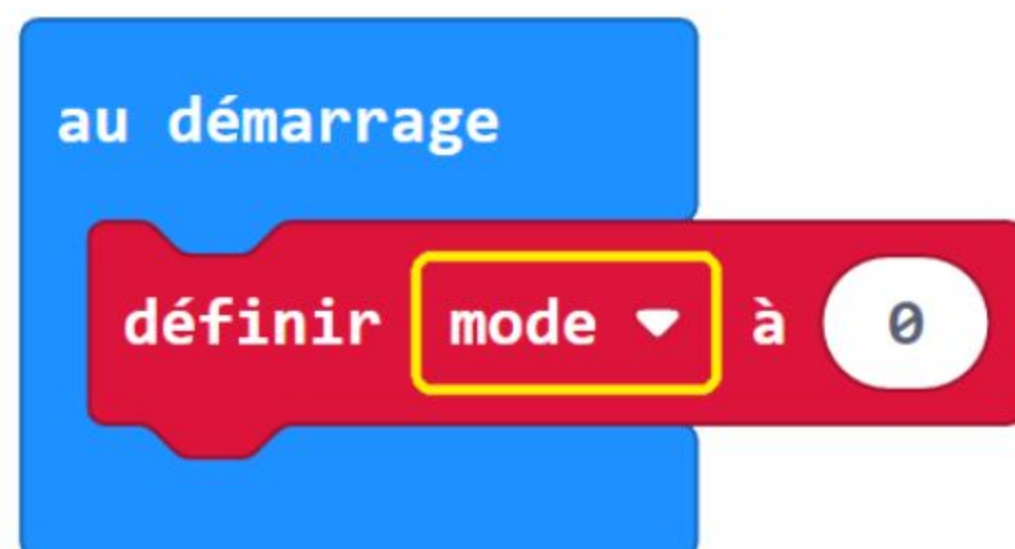
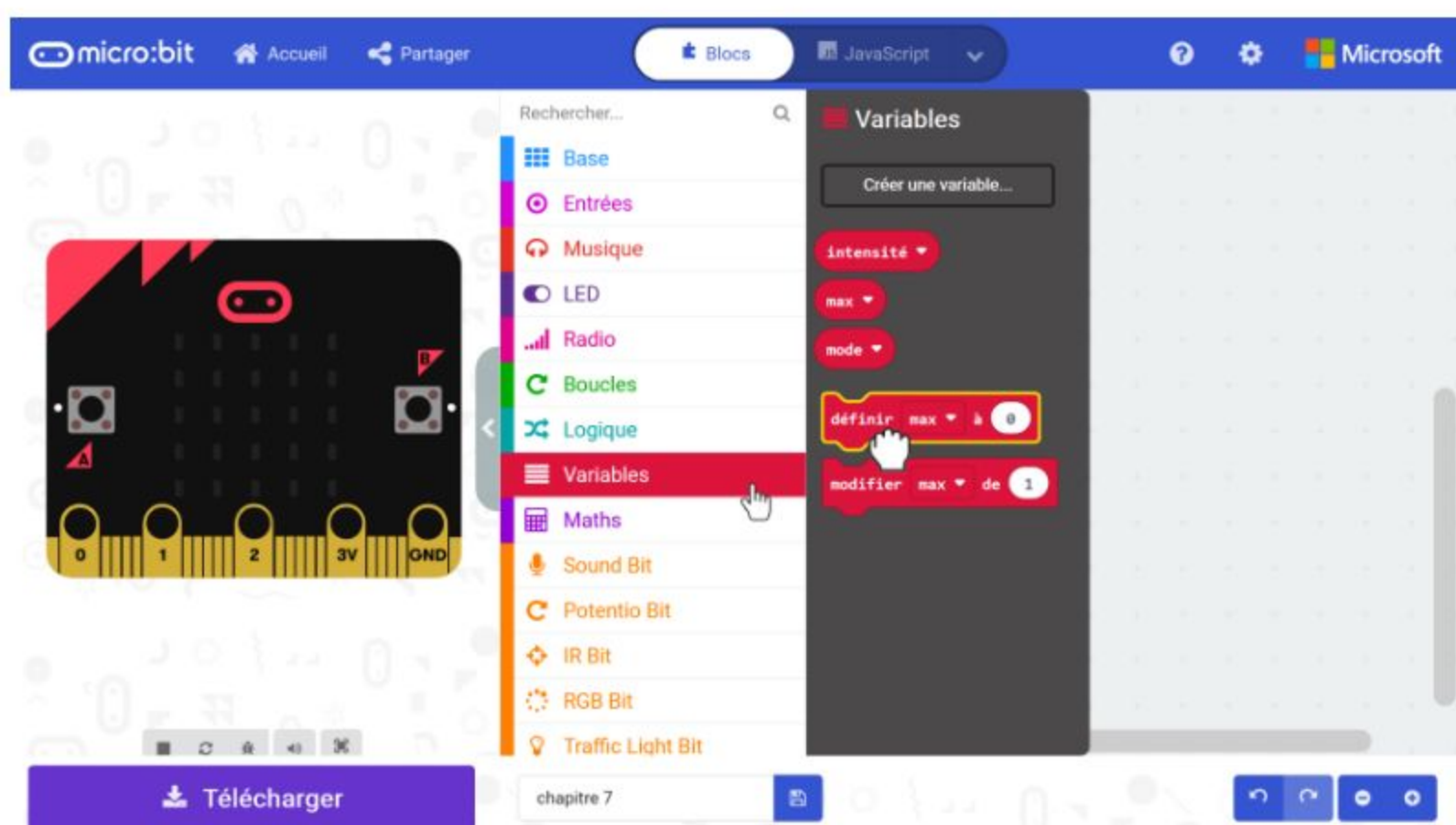


Maintenant, transformons EDU:BIT en applaudimètre.

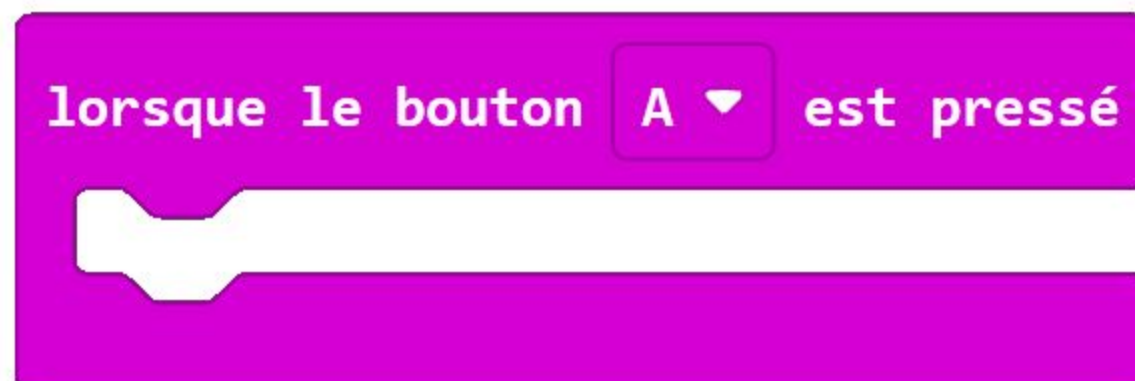
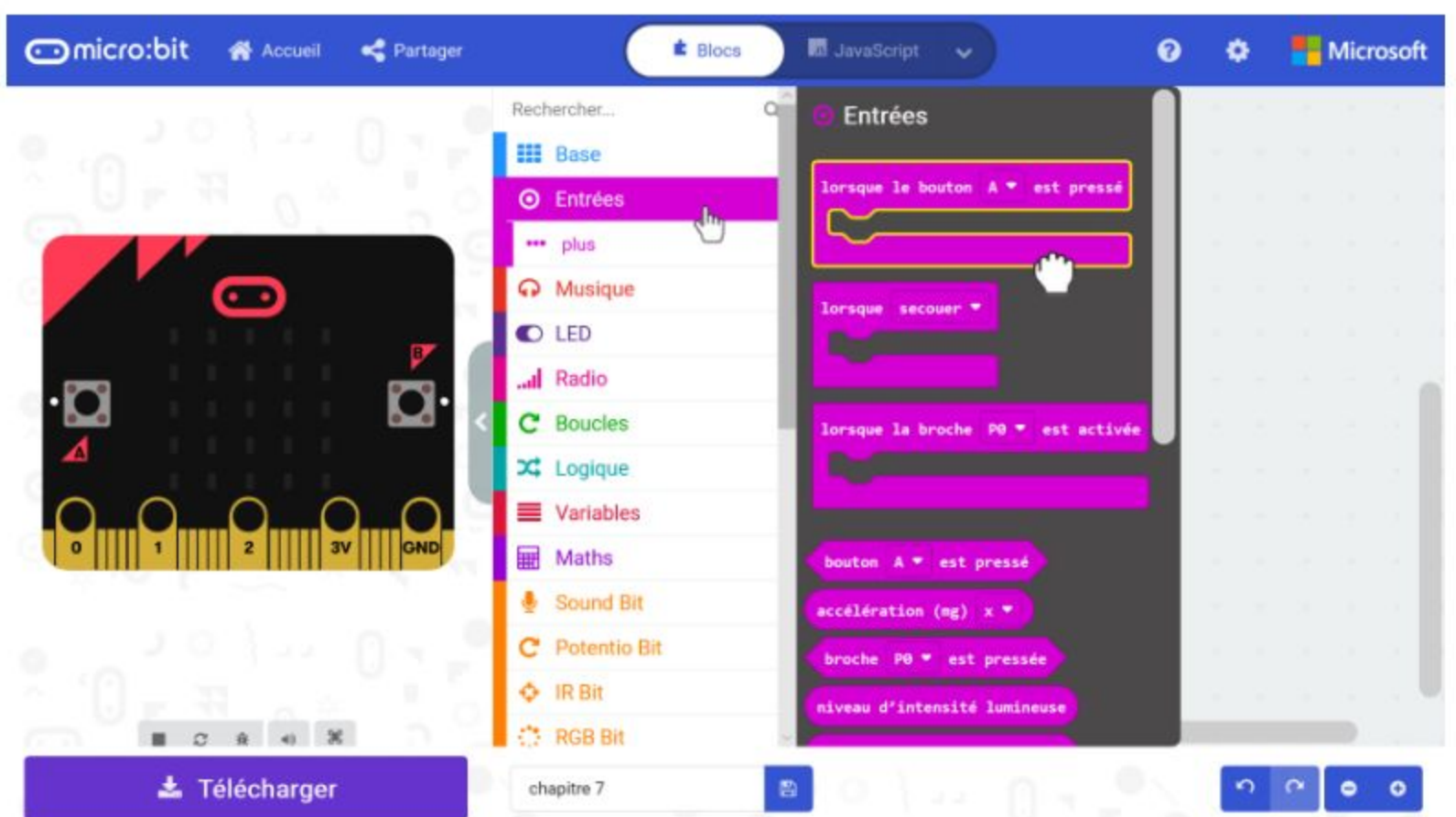
Étape 4 Démarre un nouveau projet et ajoute l'extension EDU:BIT. Clique sur la catégorie [**Variables**] puis sur [**Créer une variable...**]. Tape "mode" dans la fenêtre pop-up, puis clique sur OK. Crée deux autres variables nommées "intensité" et "max".



Étape 5 Prends un bloc [**définir _ à _**] de la catégorie [**Variables**]. Imbrique-le dans le bloc [**au démarrage**]. Choisis la variable "mode".

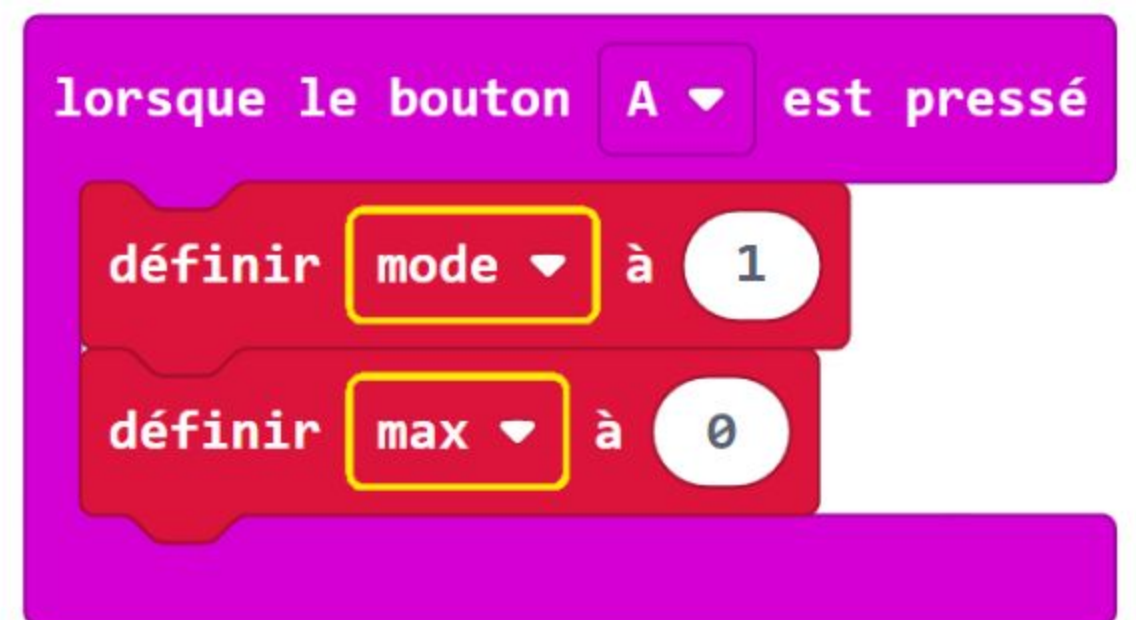
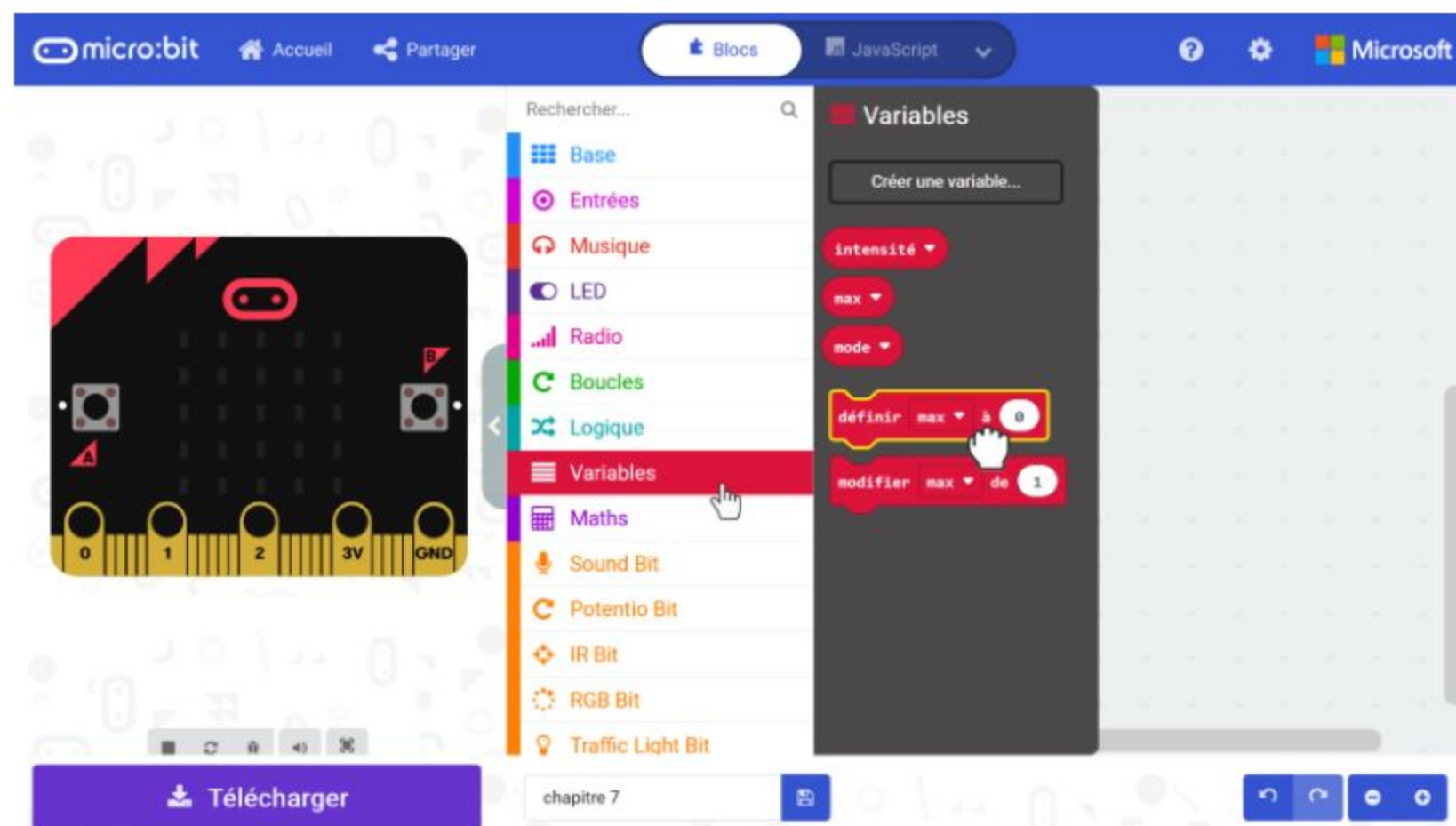


Étape 6 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**].

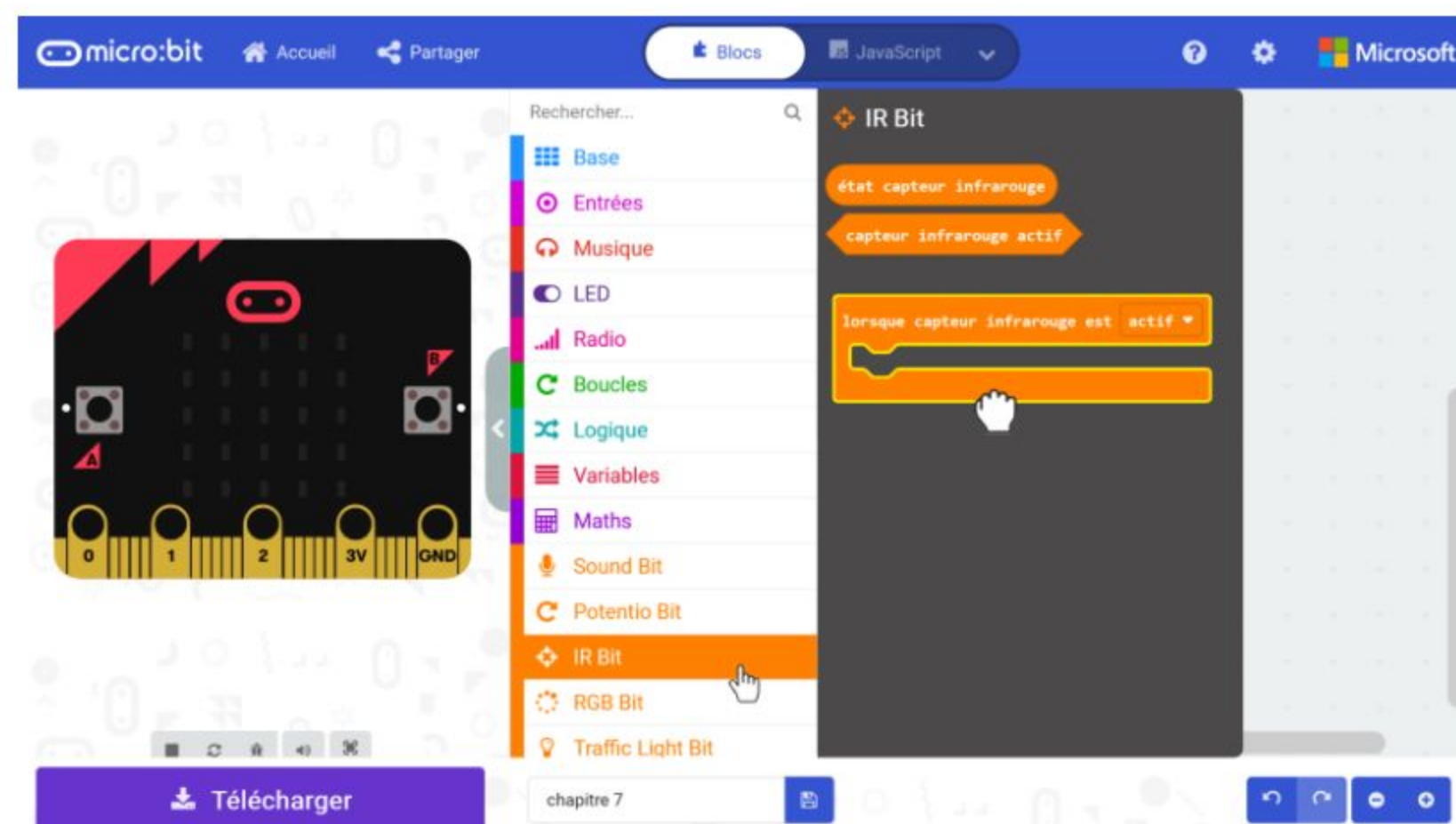




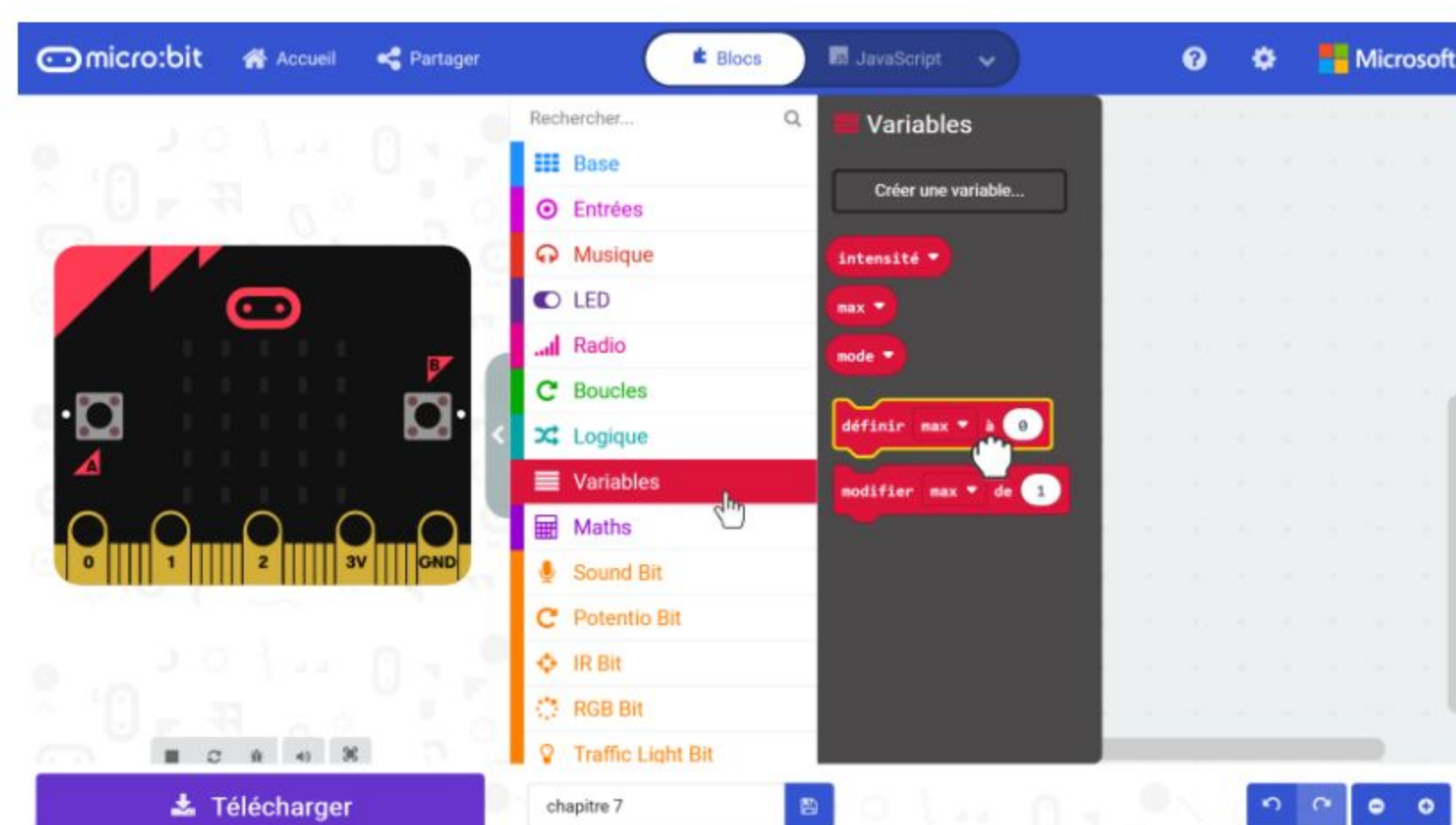
Étape 7 Prends deux blocs [définir _ à _] de la catégorie [Variables] et imbrique-les dans le bloc [lorsque le bouton A est pressé]. Sélectionne la variable “mode” et change la valeur en 1 pour le premier bloc, sélectionne la variable “max” et change la valeur en 0 pour le second.



Étape 8 Clique sur la catégorie [IR Bit] et sélectionne le bloc [lorsque capteur infrarouge est actif].

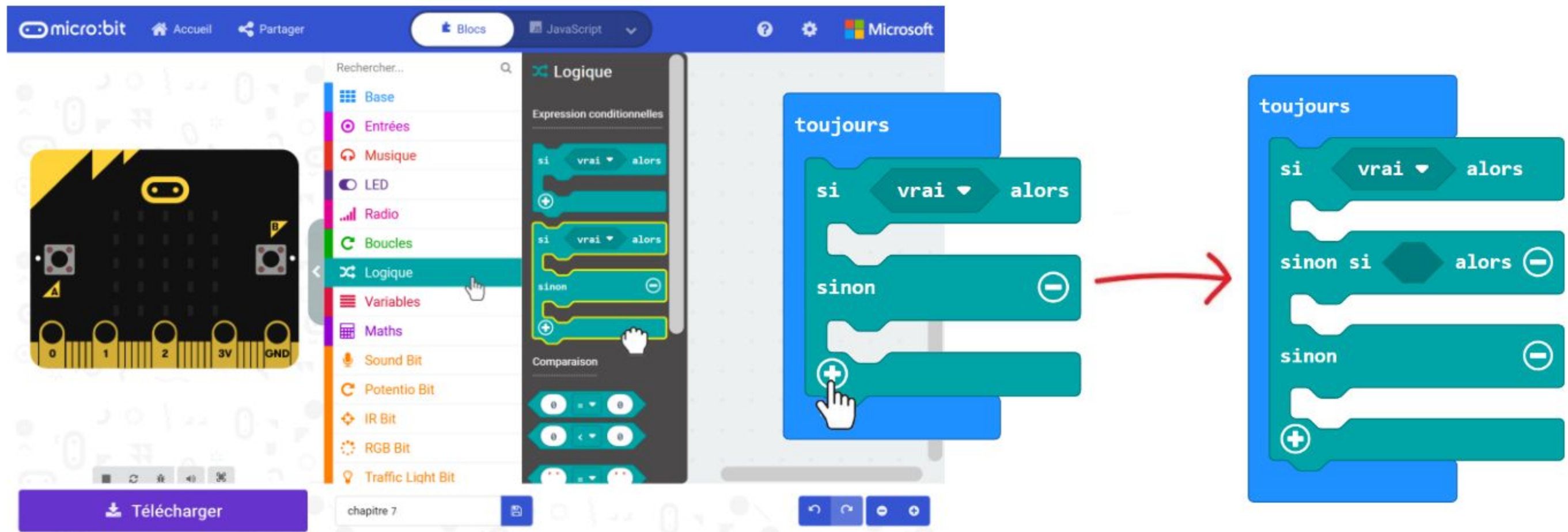


Étape 9 Clique sur la catégorie [Variables] puis sélectionne le bloc [définir _ à _]. Imbrique-le dans le bloc [lorsque capteur infrarouge est actif] et change la variable en “mode” et la valeur en 2.

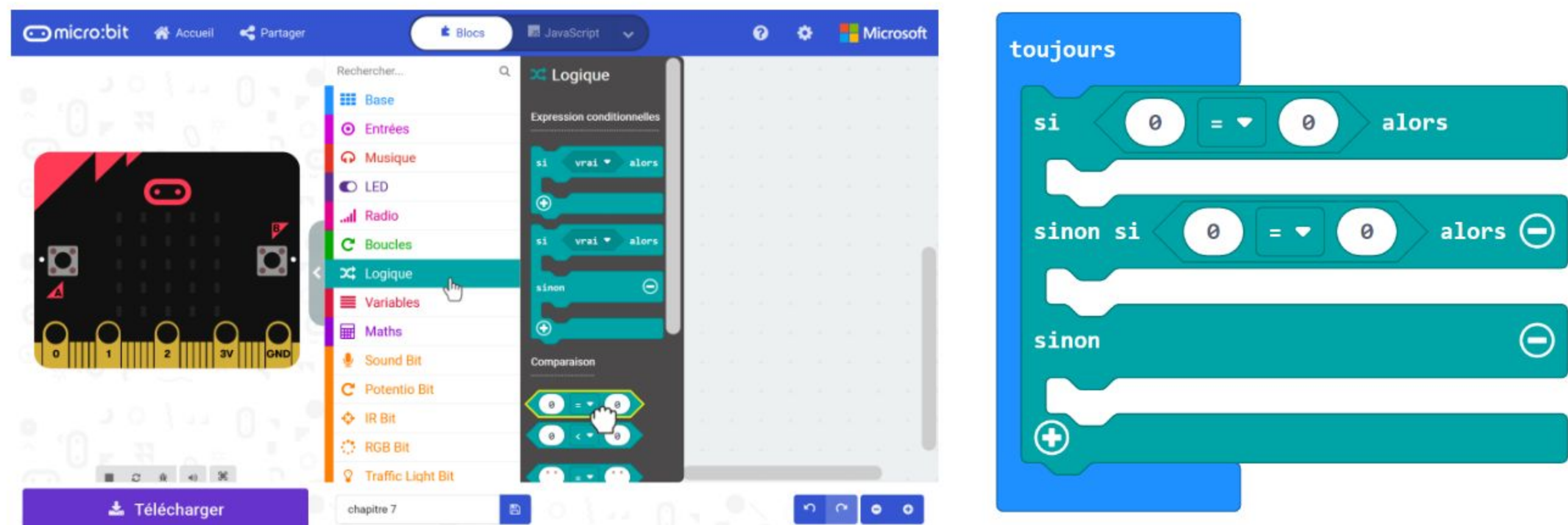


CHAPITRE 7 : Écoutons les Applaudissements !

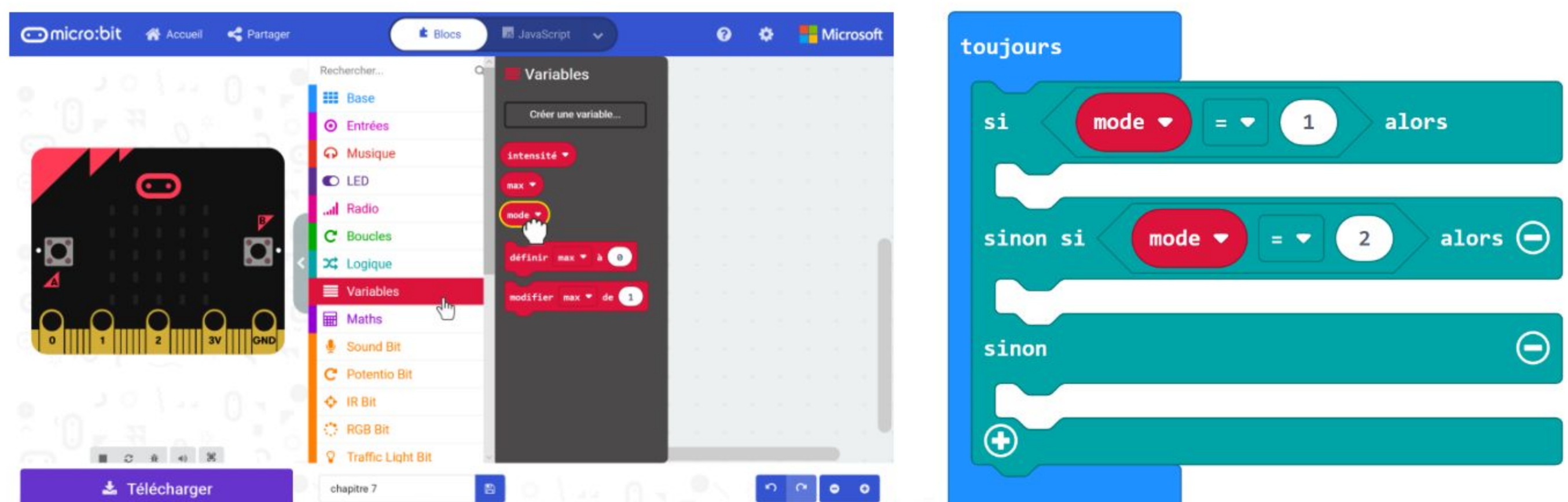
Étape 10 Clique sur la catégorie [**Logique**] et sélectionne le bloc [**si-alors-sinon**]. Imbrique-le dans le bloc [**toujours**]. Clique sur l'icône **+** pour ajouter une condition sinon-si dans le bloc.



Étape 11 Clique sur la catégorie [**Logique**] et sélectionne le bloc de comparaison [**_ = _**]. Duplique-le et insère-les dans les conditions du bloc [**si-alors-sinon**].



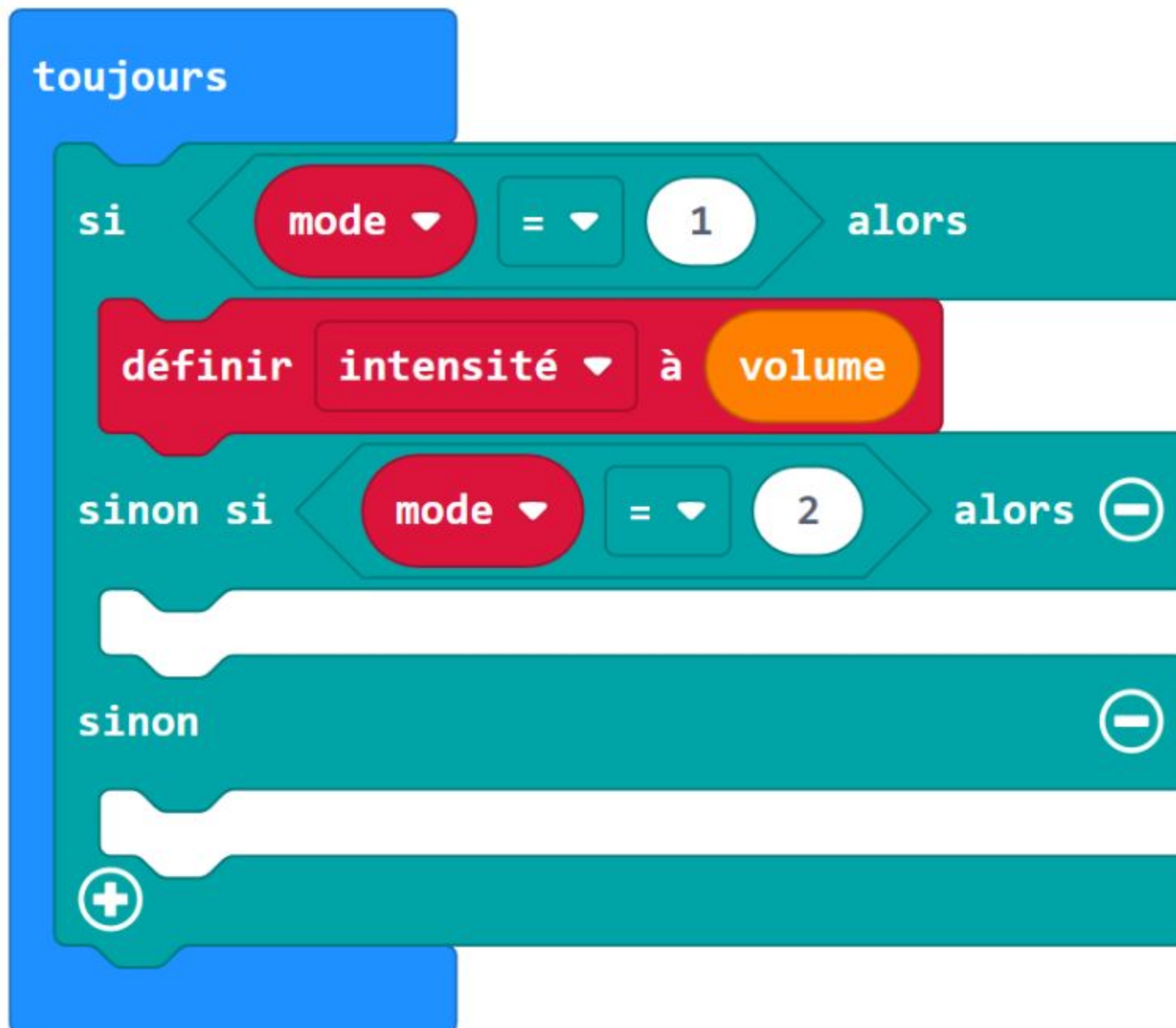
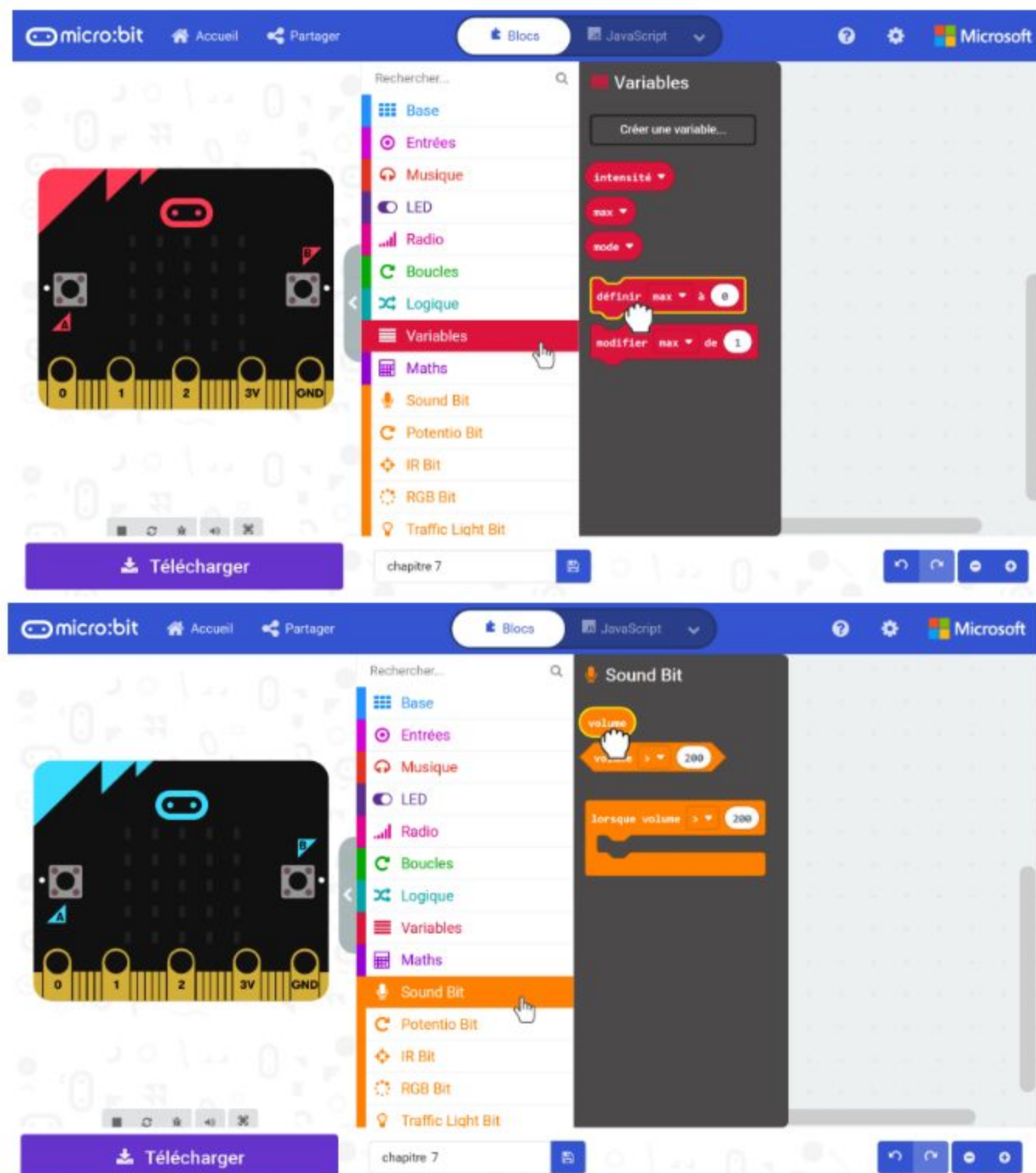
Étape 12 Prends deux blocs [**mode**] de la catégorie [**Variables**] et insère-les dans le premier membre de chaque bloc de comparaison. Saisis 1 et 2 respectivement dans les seconds membres.



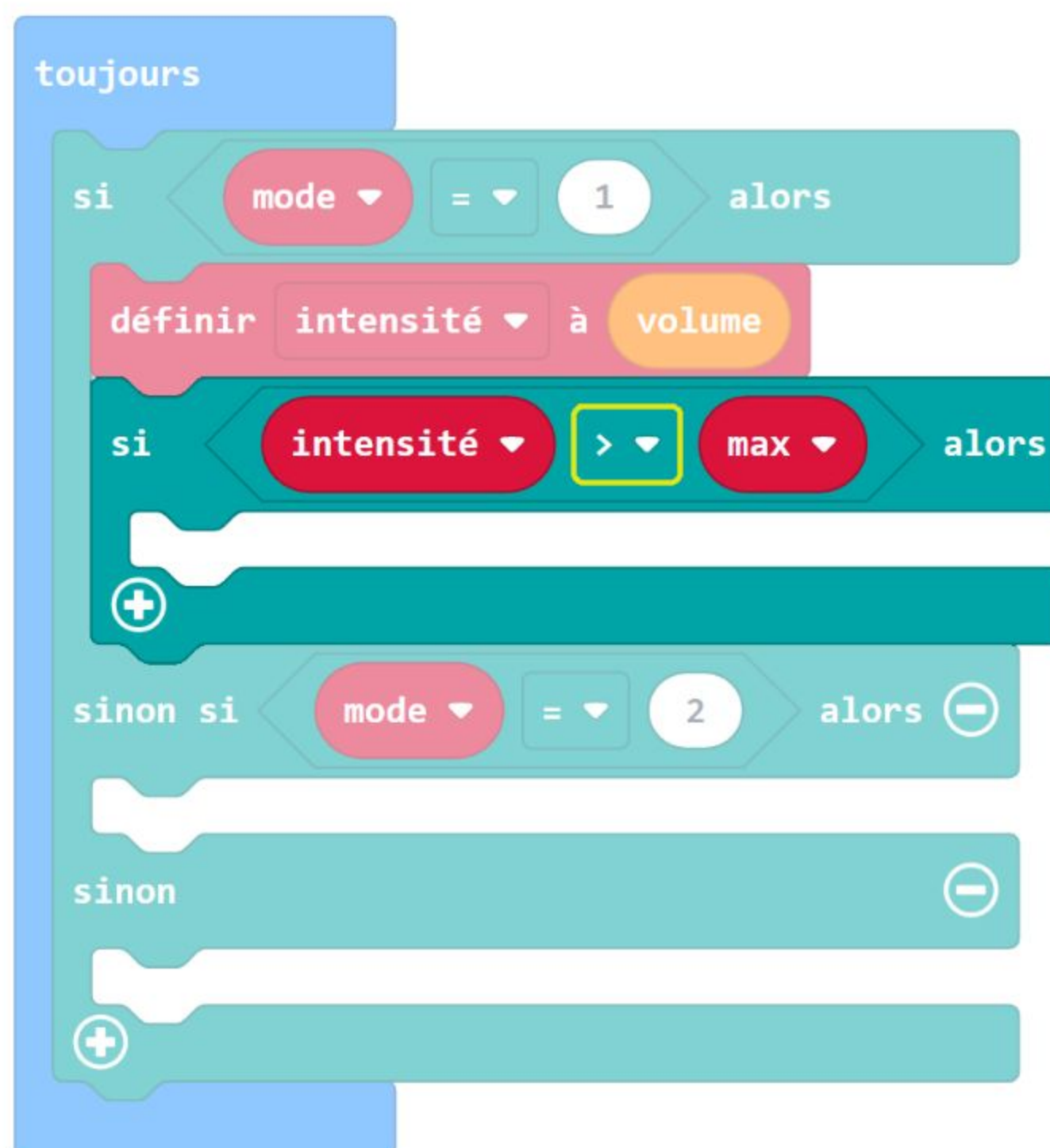
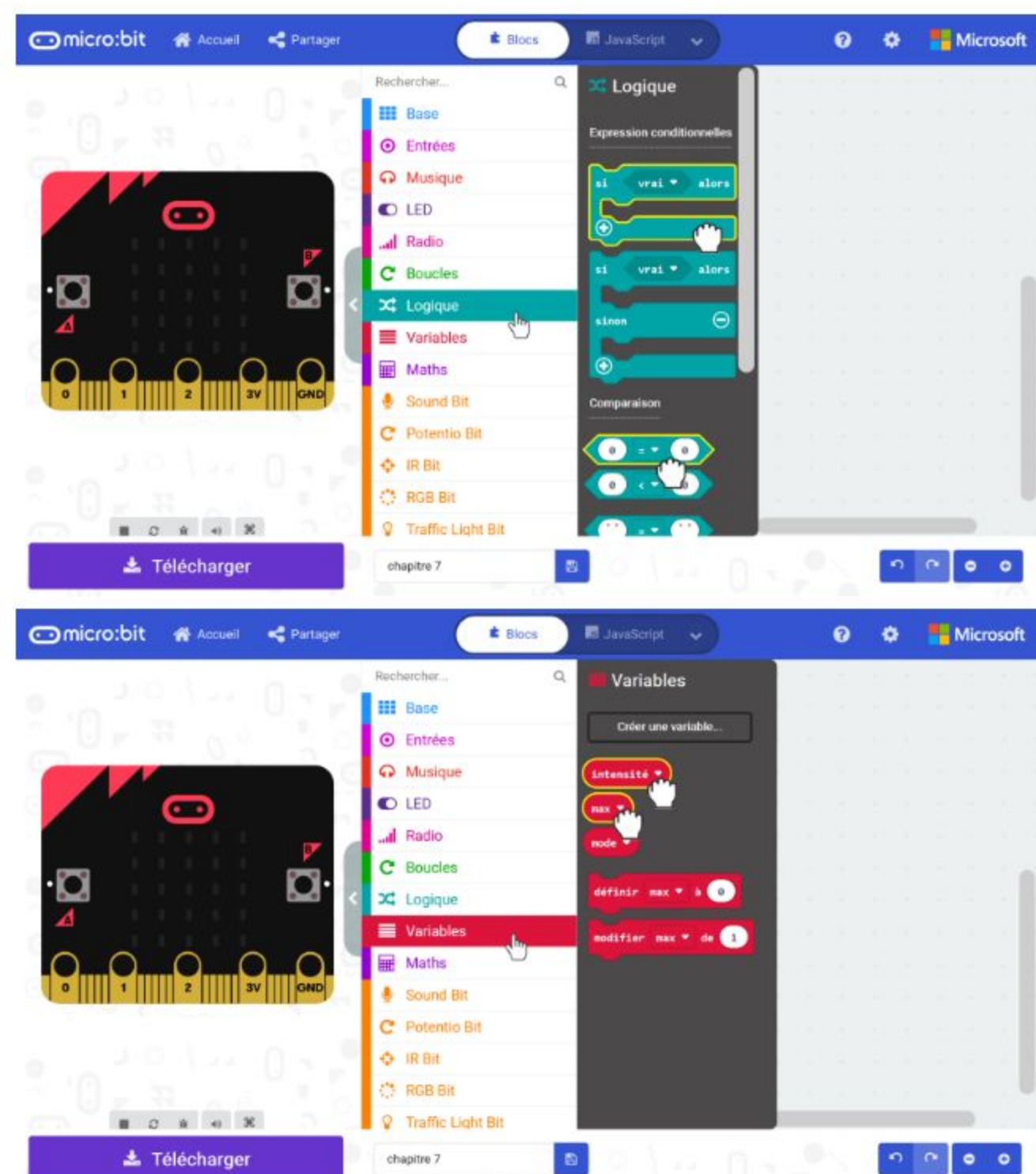
CHAPITRE 7 : Écoutons les Applaudissements !



Étape 13 Prends un bloc [définir _ à _] de la catégorie [Variables] et imbrique-le dans le bloc [si-alors-sinon]. Change la variable en "intensité" et insère le bloc [volume] de la catégorie [Sound Bit] dans le champ de valeur.

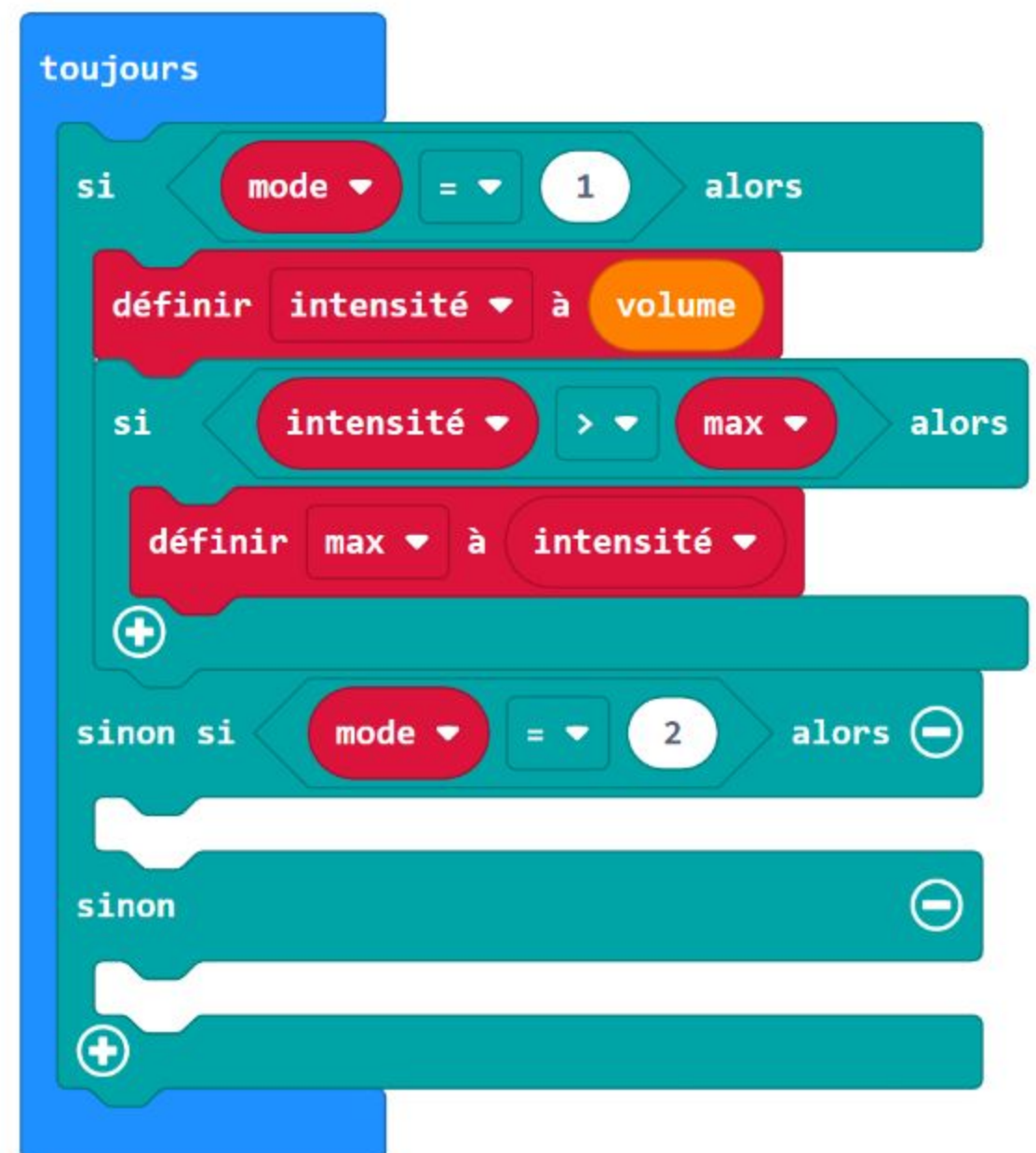
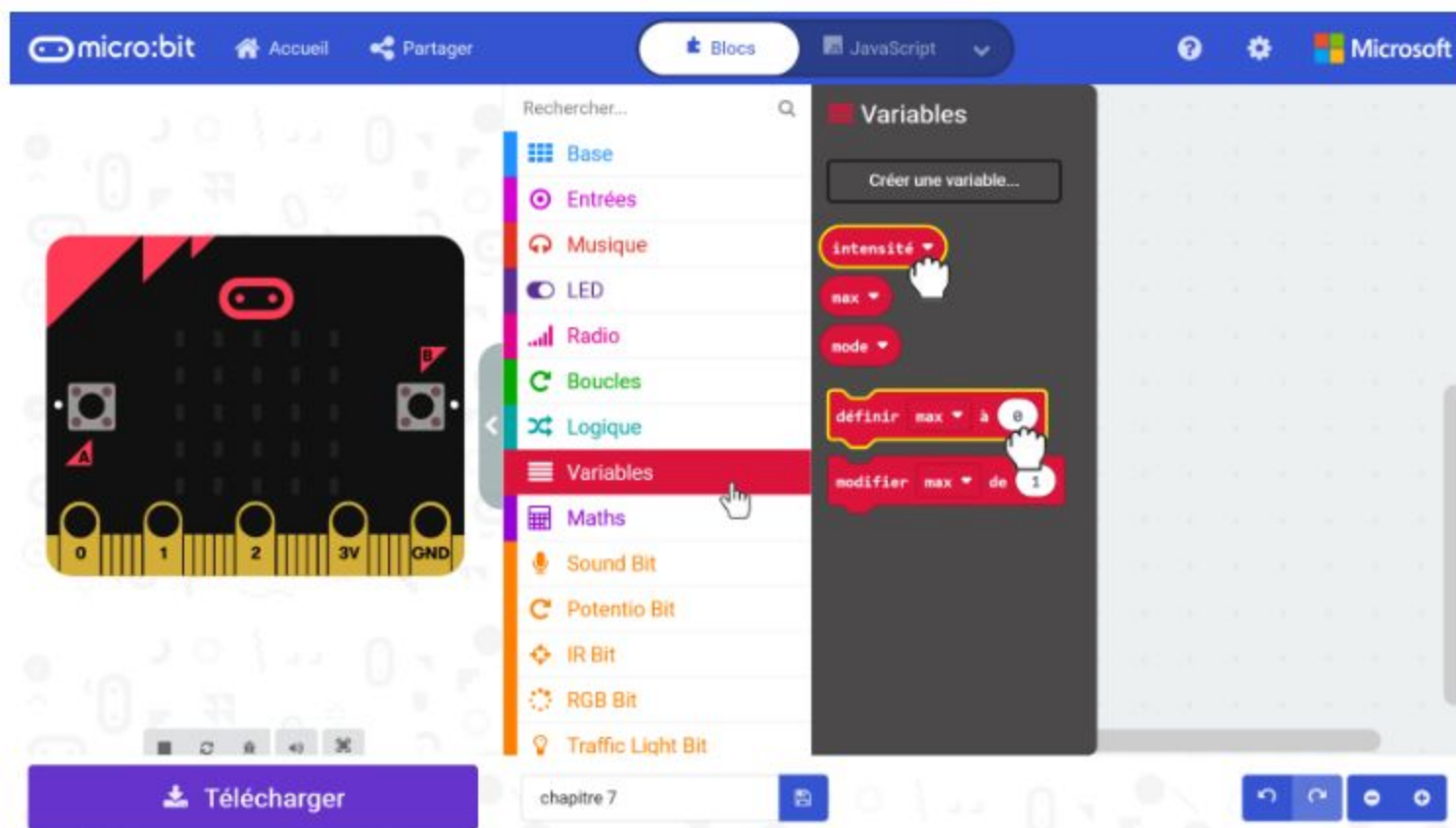


Étape 14 Clique sur la catégorie [Logique] et sélectionne le bloc [si-alors] et le bloc de comparaison [_ = _]. Change le signe "=" en ">". Insère les blocs [intensité] et [max] de la catégorie [Variables] dans les membres du bloc de comparaison.

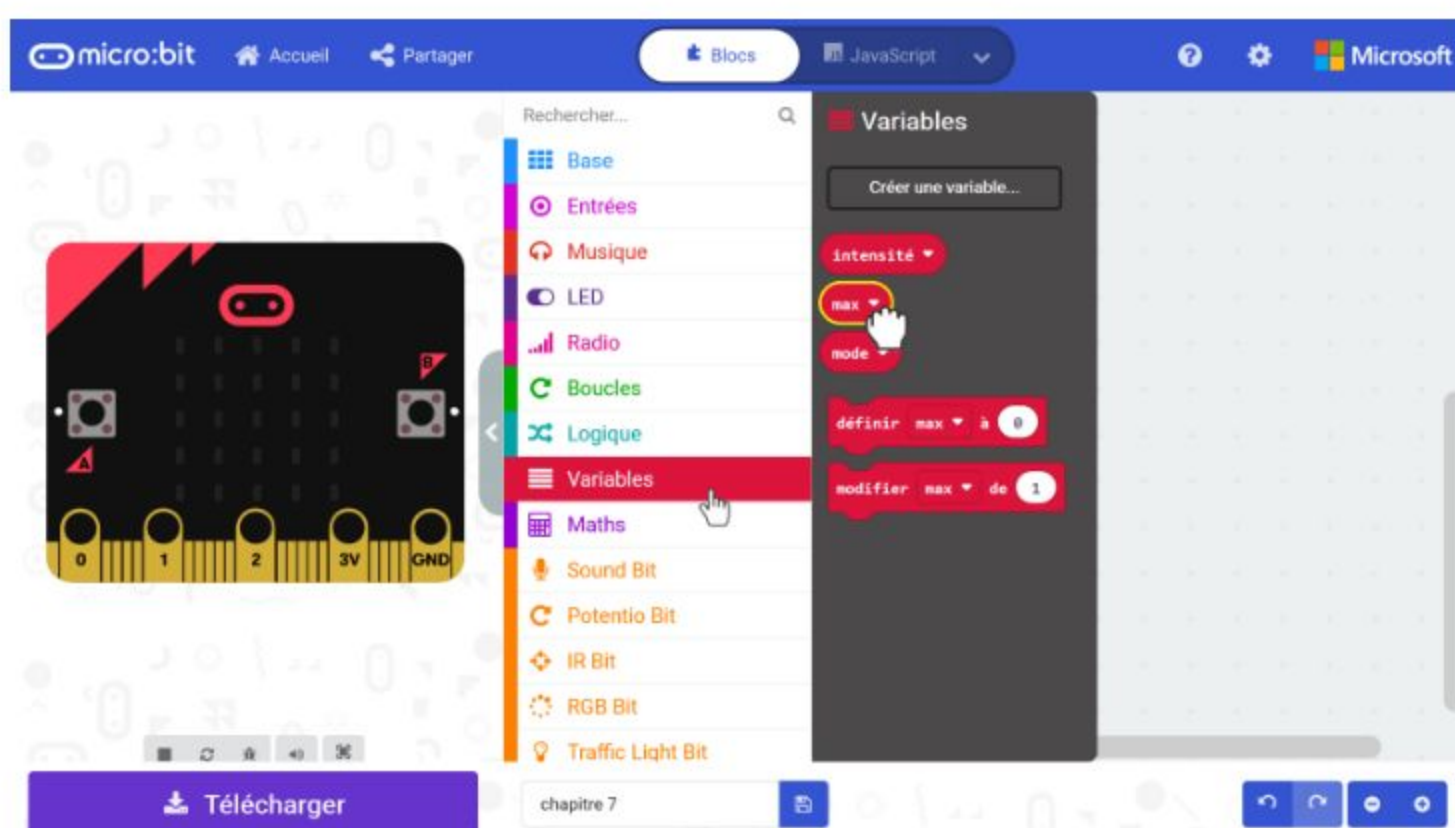
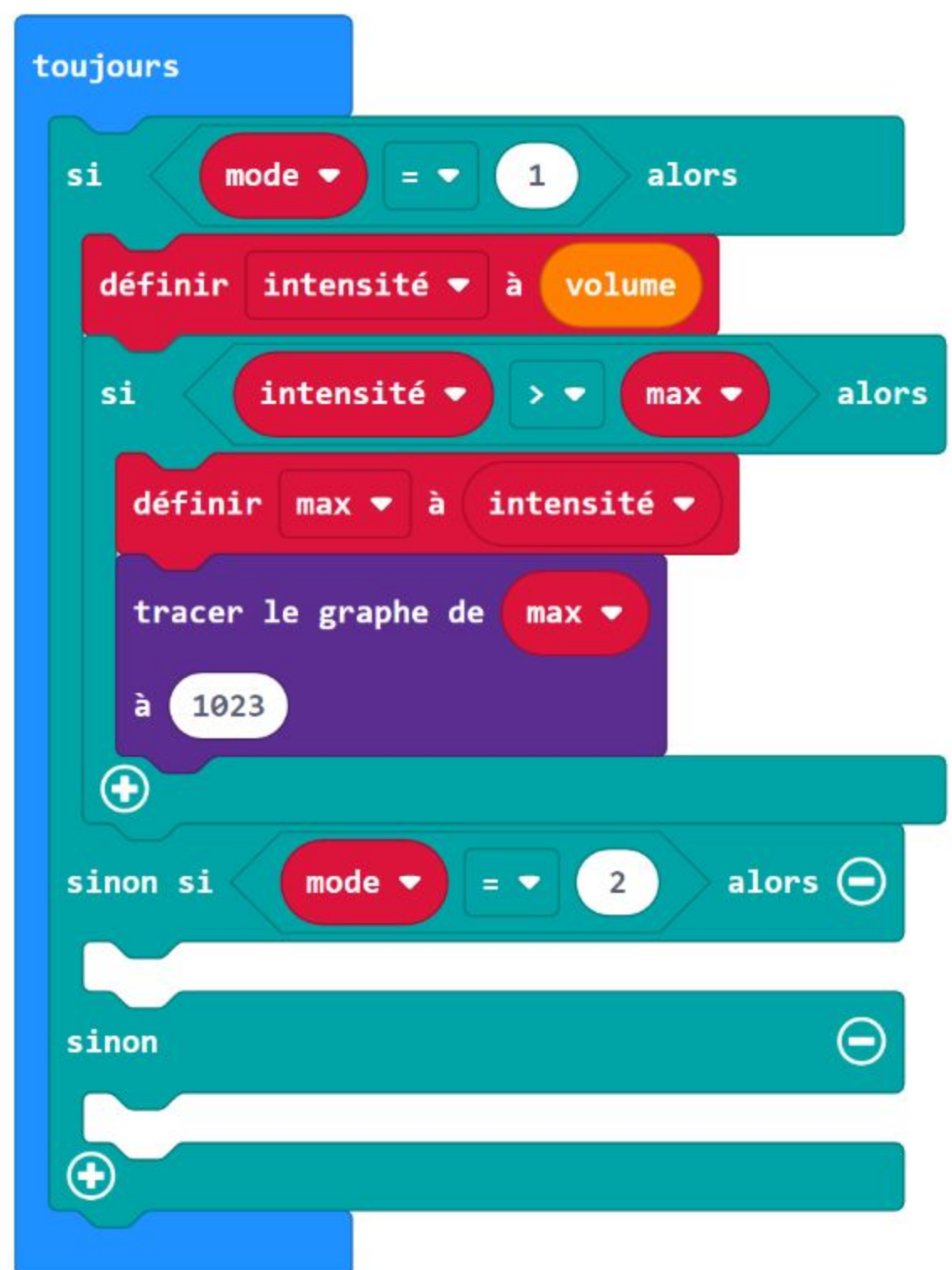
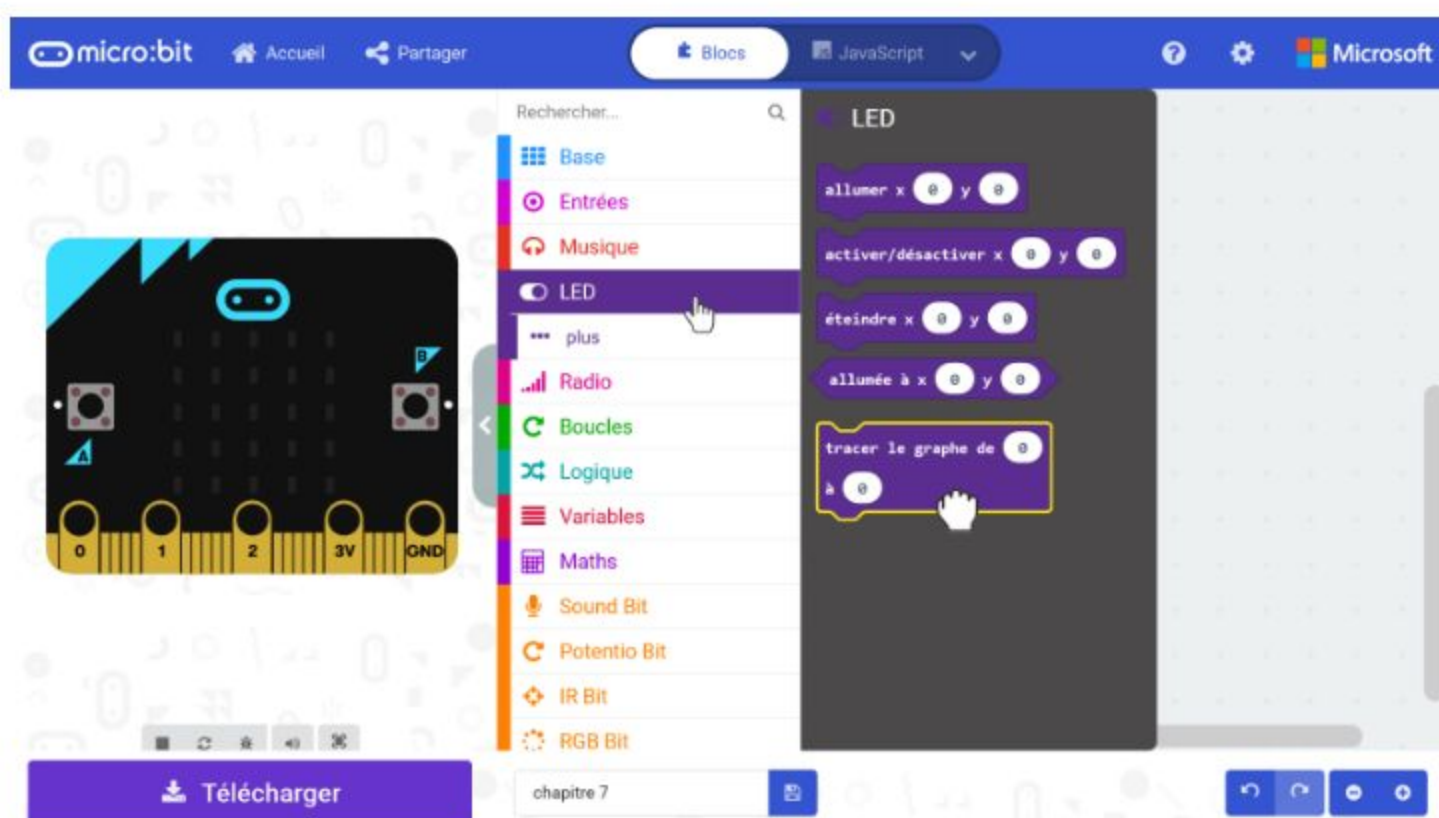


CHAPITRE 7 : Écoutons les Applaudissements !

Étape 15 Clique sur la catégorie **[Variables]** et sélectionne le bloc **[définir _ à _]**. Imbrique-le dans le bloc **[si-alors]** et insère le bloc **[intensité]** de la catégorie **[Variables]** dans le champ de valeur.

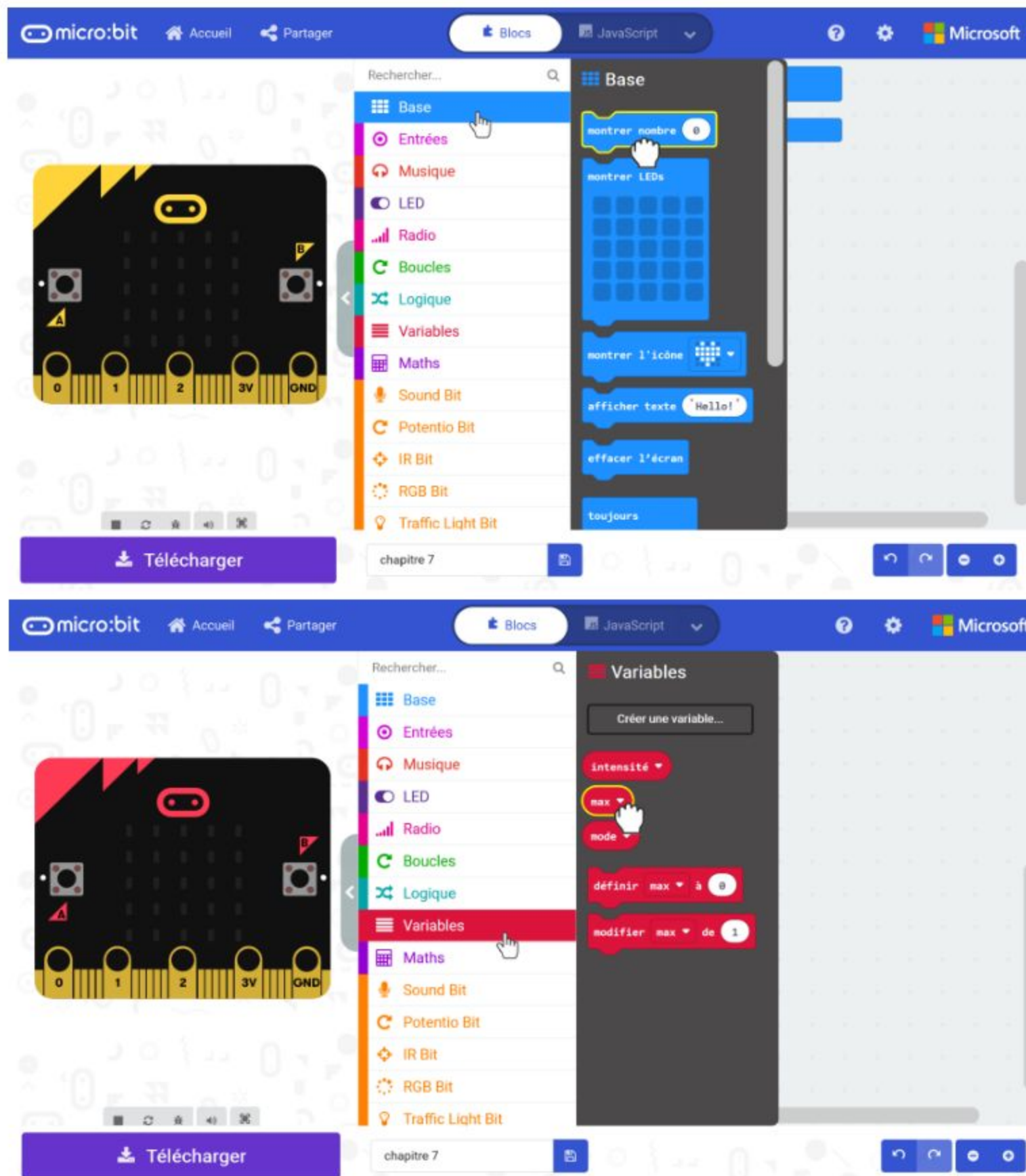


Étape 16 Clique sur la catégorie **[LED]** et sélectionne le bloc **[tracer le graphe de _ à _]**. Clique sur la catégorie **[Variables]** et sélectionne le bloc **[max]**. Imbrique-le dans le bloc **[tracer le graphe de _ à _]** et change la valeur en **1 023**.





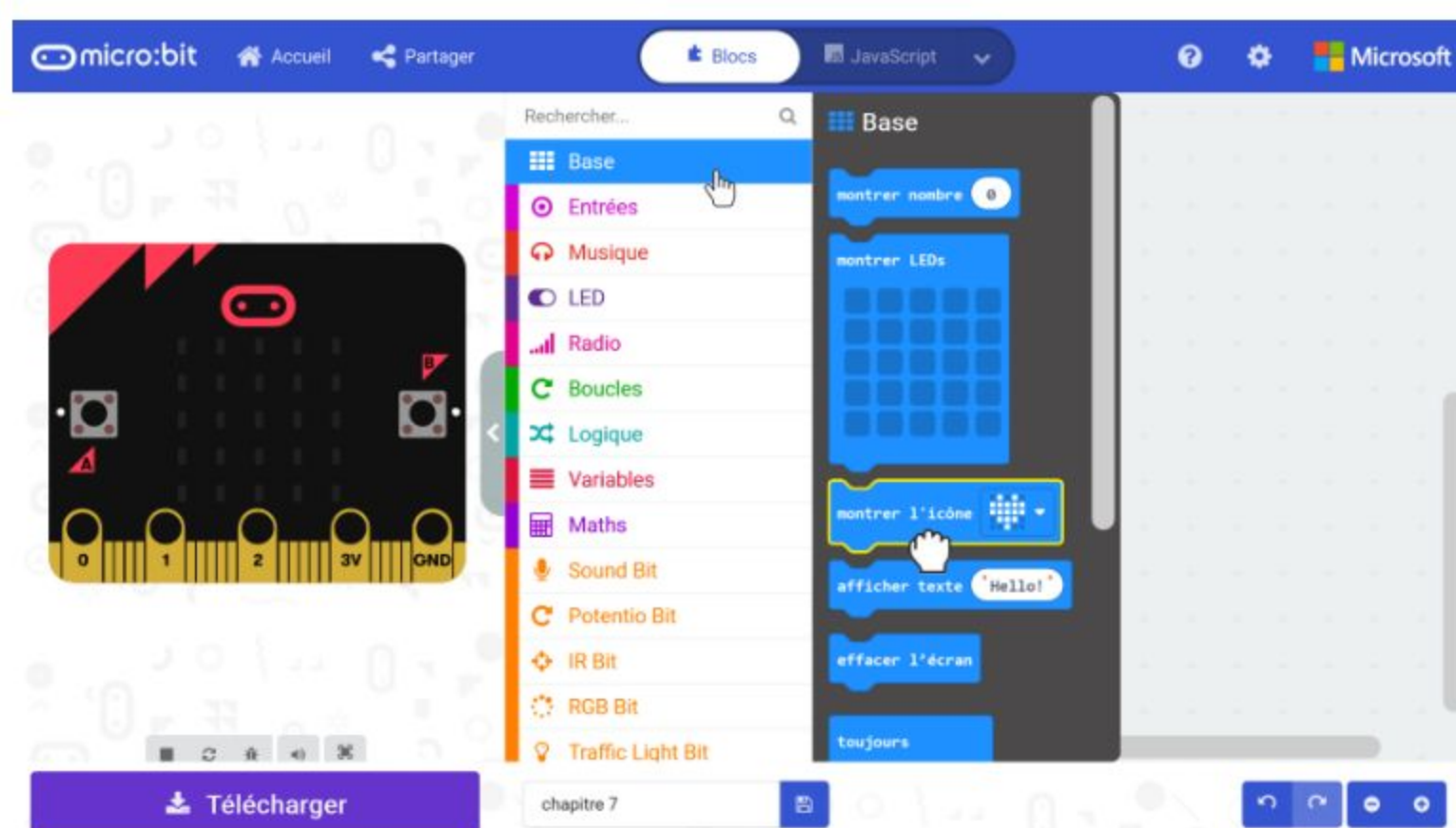
Étape 17 Cliquez sur la catégorie [Base] et sélectionnez le bloc [montrer nombre]. Imbriquez-le dans le deuxième emplacement du bloc [si-alors-sinon]. Prenez un bloc [max] de la catégorie [Variables] et insérez-le dans le champ de valeur du bloc [montrer nombre].



```

toujours
si mode = 1 alors
  définir intensité à volume
  si intensité > max alors
    définir max à intensité
    tracer le graphe de max
    à 1023
  sinon si mode = 2 alors
    montrer nombre max
  sinon
  
```

Étape 18 Cliquez sur la catégorie [Base] et sélectionnez le bloc [montrer l'icône]. Imbriquez-le dans le dernier emplacement du bloc [si-alors-sinon].



```

toujours
si mode = 1 alors
  définir intensité à volume
  si intensité > max alors
    définir max à intensité
    tracer le graphe de max
    à 1023
  sinon si mode = 2 alors
    montrer nombre max
  sinon
    montrer l'icône
  
```

CHAPITRE 7 : Écoutons les Applaudissements !

Voici le code complet :

au démarrage
Au démarrage, on initialise mode à 0.

lorsque le bouton A est pressé
Lorsque le bouton A est pressé, mode prend la valeur 1, max prend la valeur 0.

lorsque capteur infrarouge est actif
Lorsque le capteur infrarouge est actif, mode prend la valeur 2.

toujours
Vérifie constamment la valeur de mode.

si mode = 1 alors
définir intensité à volume
si intensité > max alors
définir max à intensité
tracer le graphe de max à 1023

sinon si mode = 2 alors
montrer nombre max

sinon
montrer l'icône [LED grid icon]

Si "mode" = 2 (ie : capteur infrarouge déclenché), alors affiche la valeur actuelle de la variable max, correspondant à la valeur du niveau de bruit la plus haute relevée.

Si "mode" n'est égal ni à 1 ni à 2, alors afficher l'icône cœur.

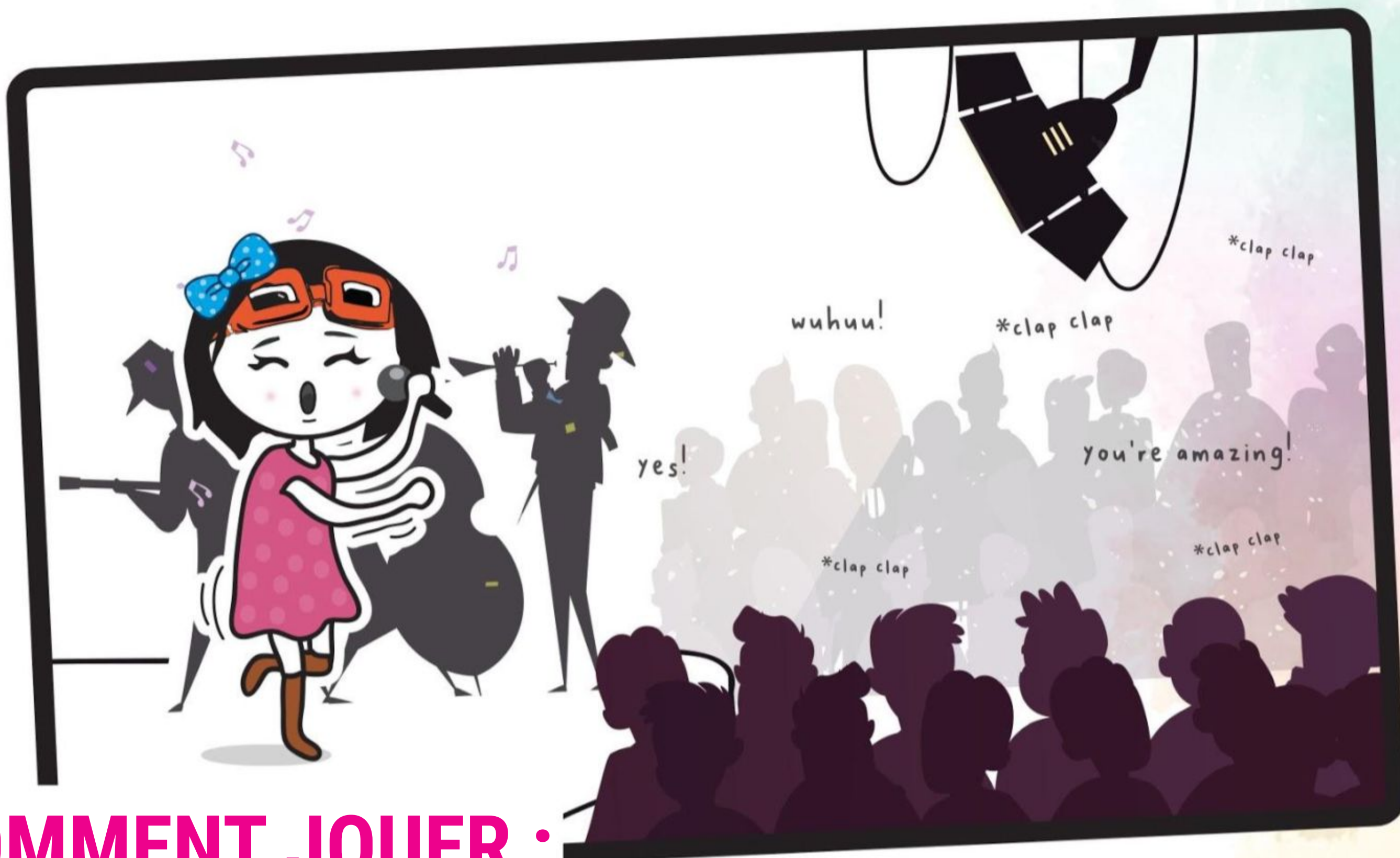
Essayons !



Étape 19 Flashe le code sur EDU:BIT et tu auras un applaudimètre prêt à être utilisé pour le jeu Incroyable talent.

Jouons

Écoutons les Applaudissements !



COMMENT JOUER :

- Les participants prennent le temps de préparer une courte performance, individuelle ou collective. Ils peuvent choisir de chanter, danser ou même raconter des histoires drôles.
- Lorsque tout le monde est prêt, place à la démonstration de talents. Après chaque performance, le public évalue la performance en applaudissant. Plus ils ont apprécié le spectacle, plus les spectateurs tapent fort dans les mains.
- Une fois que les applaudissements cessent, déclencher le capteur infrarouge pour qu'il affiche le score (le pic de bruit enregistré).
- Penser à appuyer sur le bouton A pour remettre le score à 0 pour la performance suivante.
- La performance ayant récolté les applaudissements les plus forts remporte le concours. Amuse-toi !

DÉCHIFFRE

LE CODE

Lorsque nous avons de multiples tâches dans un programme, nous pouvons utiliser le déclenchement d'événements pour passer de l'une à l'autre. Dans ce programme, pour commencer doucement, nous avons utilisé le bloc toujours pour constamment vérifier la valeur de la variable mode puis exécuté le code du bloc correspondant.

Initialise mode à 0
(ie : en attente)
au démarrage.

```
au démarrage
  définir mode à 0

lorsque le bouton A est pressé
  définir mode à 1
  définir max à 0

lorsque capteur infrarouge est actif
  définir mode à 2
```

Blocs événements qui, lorsqu'ils sont exécutés, permettent de passer d'un mode à l'autre.

Vérifie constamment quelle est la valeur actuelle de mode, puis exécute les tâches (blocs de code) assignées à ce mode spécifique.

```
toujours
  si mode = 1 alors
    définir intensité à volume
    si intensité > max alors
      définir max à intensité
      tracer le graphe de max
      à 1023
    sinon si mode = 2 alors
      montrer nombre max
    sinon
      montrer l'icône [icône]
```



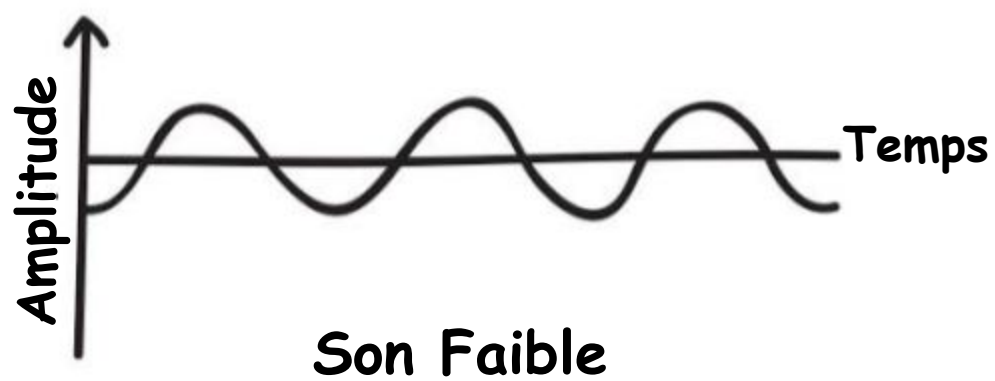
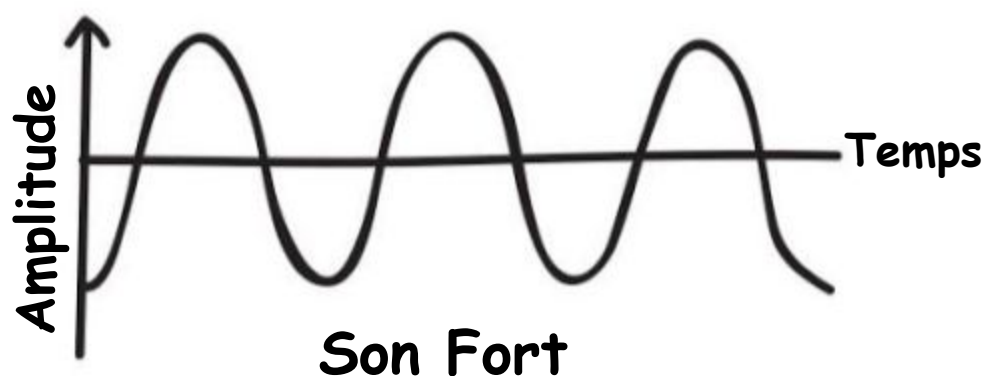
Si tu as d'autres tâches / modes à ajouter, tu peux toujours ajouter dans ton code d'autres blocs déclenchés par un événement comme [lorsque secouer] et [lorsque volume > _], et cliquer sur le bouton ⊕ du bloc [si-alors-sinon] pour ajouter d'autres conditions.

FAIT AMUSANT !



Un son est produit lorsqu'un objet vibre, par exemple une batterie, lorsque le batteur frappe la peau d'un tom. Cette vibration se propage en faisant vibrer à leur tour les molécules d'air, ce qui crée une onde sonore.

Un **capteur de son** est un module qui détecte l'intensité d'une onde sonore (ie : le volume sonore) et la convertit en signal électrique.



En un sens, un capteur de son fonctionne comme nos oreilles, qui convertissent les vibrations de l'air en un signal électrochimique que notre cerveau peut traduire en un son que nous comprenons.



Voici quelques applications communes :

- Alarme anti-cambriolage qui détecte le son
- Interrupteur sonore pour allumer la lumière
- Babyphone audio

Penses-tu qu'un capteur de son peut détecter du bruit dans l'espace ? Pourquoi ? Ou pourquoi pas ?



APPLICATION DÉFI

Programme EDU:BIT pour qu'il serve de détecteur de bruit. Allume une LED du Traffic Light Bit pour indiquer le niveau de bruit.

Niveau de bruit	Plage de valeurs du volume sonore	Traffic Light Bit
Trop bruyant ; il faut baisser d'un ton.	de () à 1 023	LED rouge
Légèrement bruyant ; faire attention au volume.	de () à ()	LED jaune
Niveau de bruit acceptable. Bien !	de 0 à ()	LED verte

Voici quelques conseils pour toi...

#1 Tu dois d'abord déterminer les valeurs seuils pour chaque niveau sonore.

#2 Pour une meilleure stabilité des résultats, calcule la moyenne des volumes sur des intervalles de temps réguliers.



Trop facile ? Essayons de relever un peu le niveau du challenge. Modifie ton code pour que les valeurs seuils soient définies par le potentiomètre (Potentio Bit).



CHAPITRE 8

Allons Faire un Tour !

Moteur DC

Let's Play / Chapter 8

EDU:BIT TWISTER



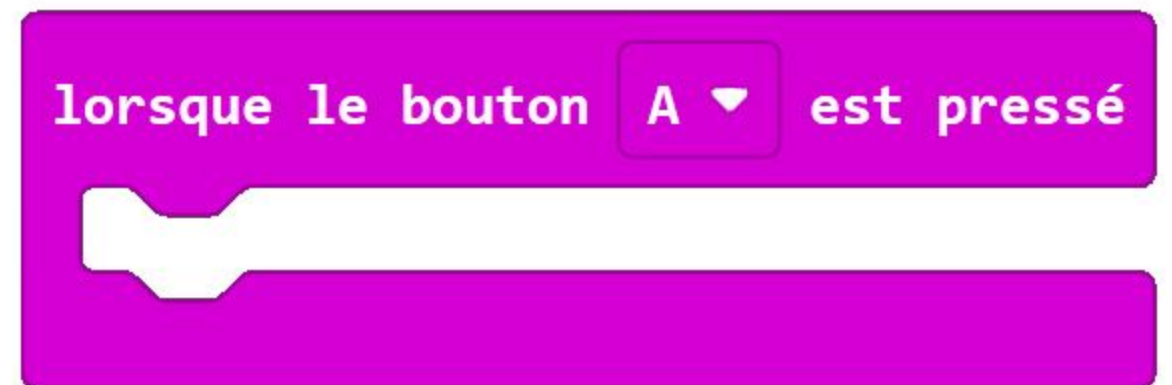
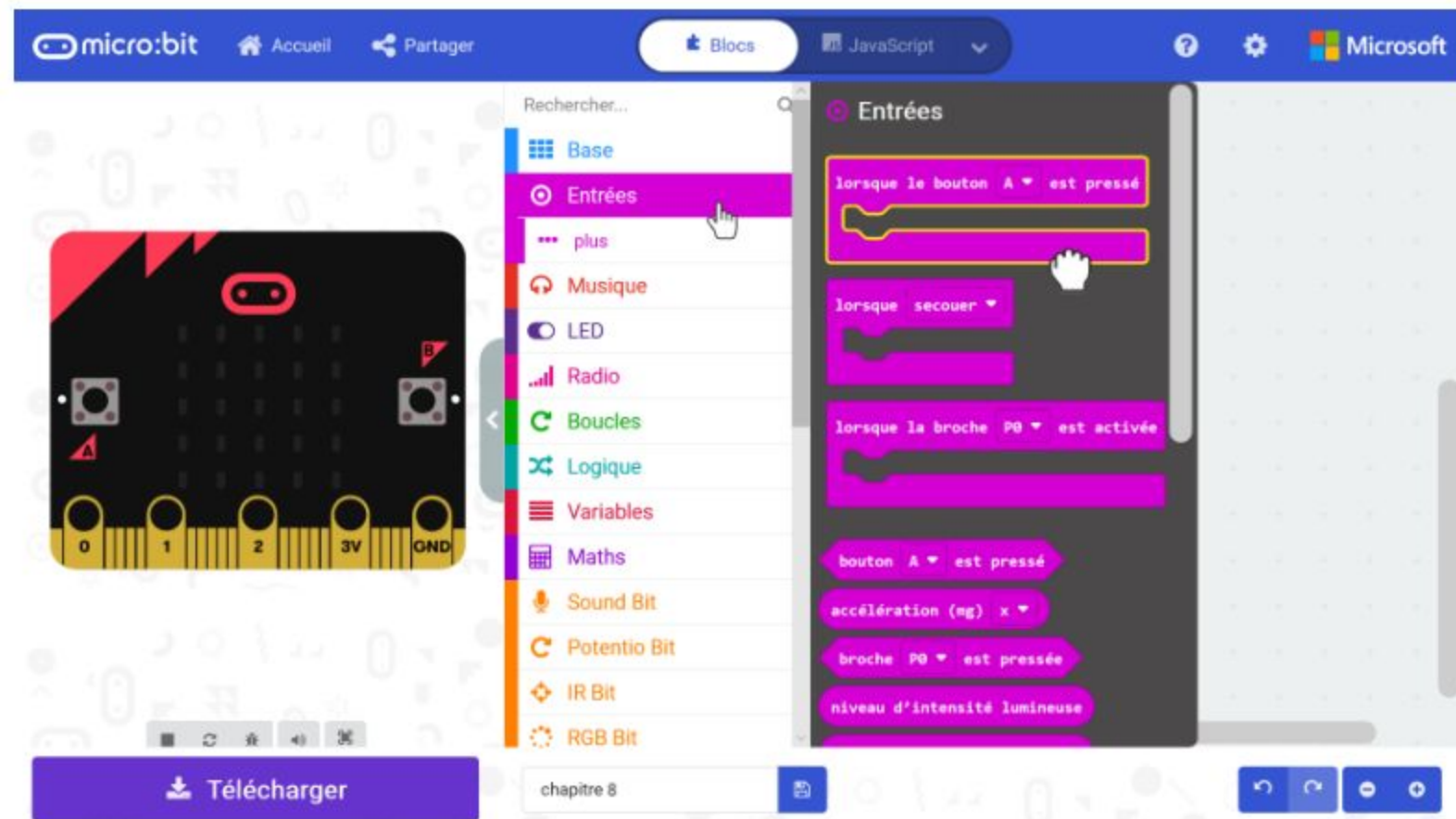
link.cytron.io/edubit-chapter-8

EDU Training & Proj

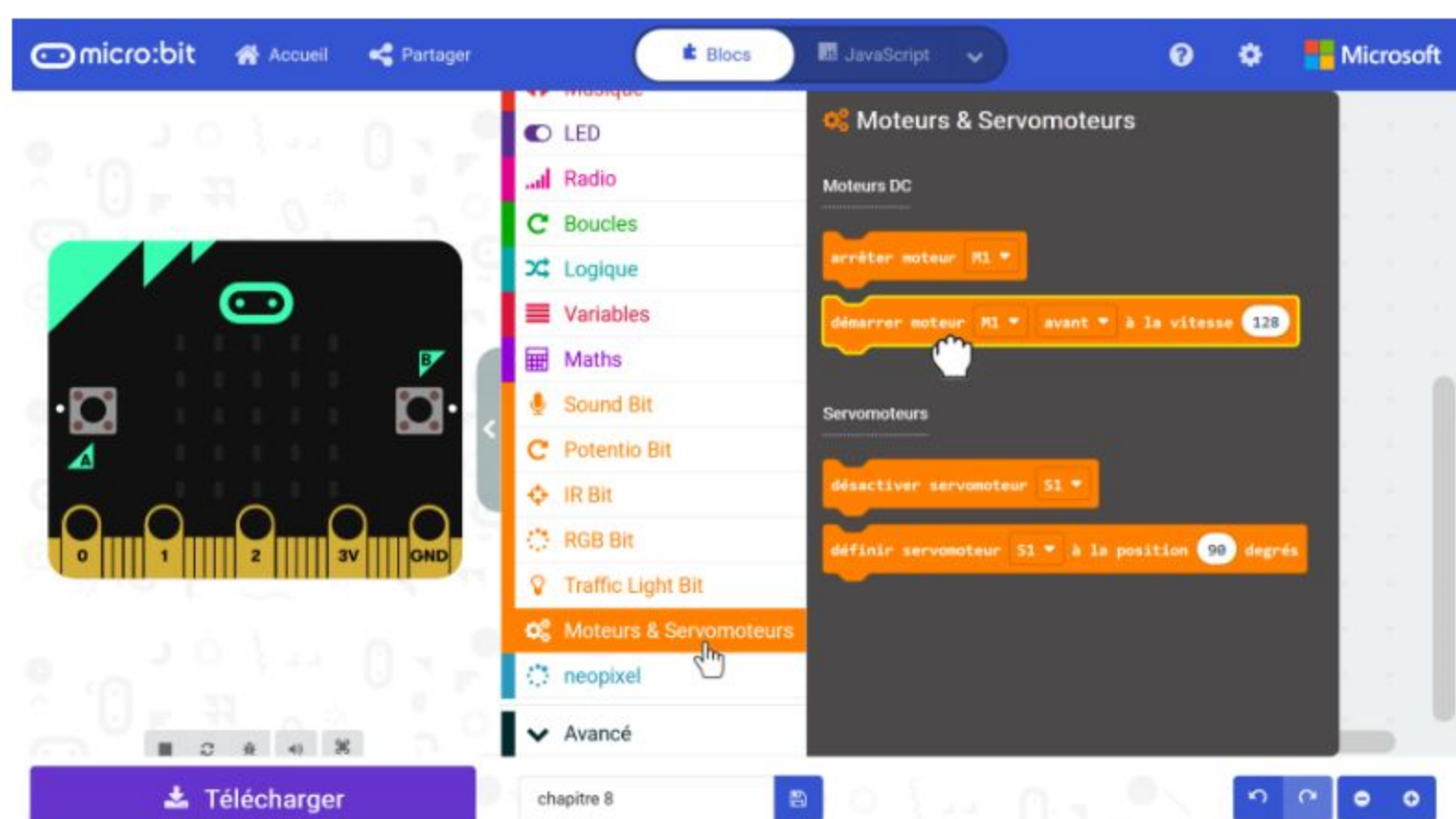
www.cytron.io

CODONS !

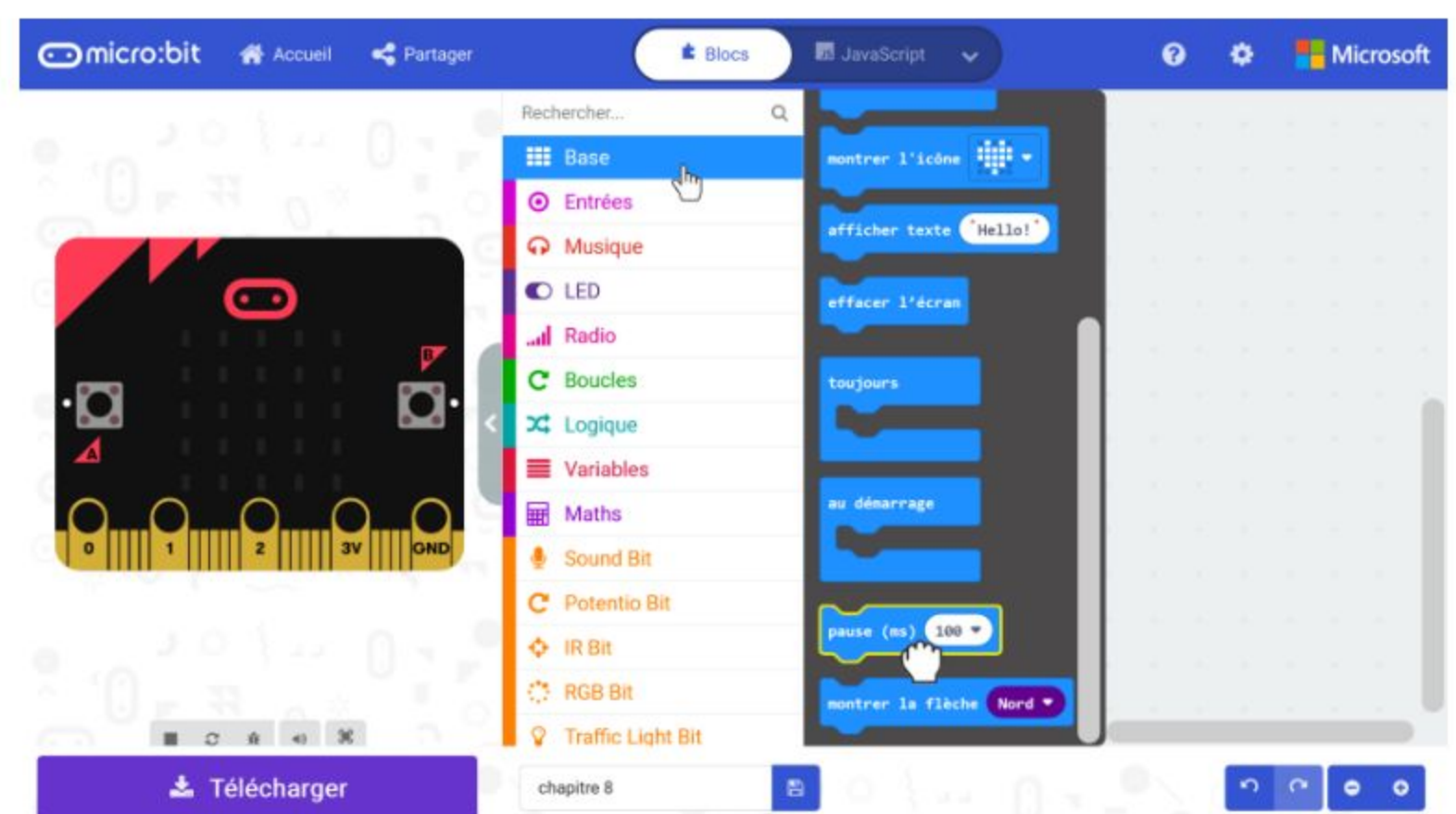
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT (voir page 40). Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**].



Étape 2 Clique sur la catégorie [**Moteurs & Servomoteurs**] et ajoute les blocs [**démarrer moteur _ à la vitesse _**] et [**arrêter moteur _**]. Change la vitesse en **80**.

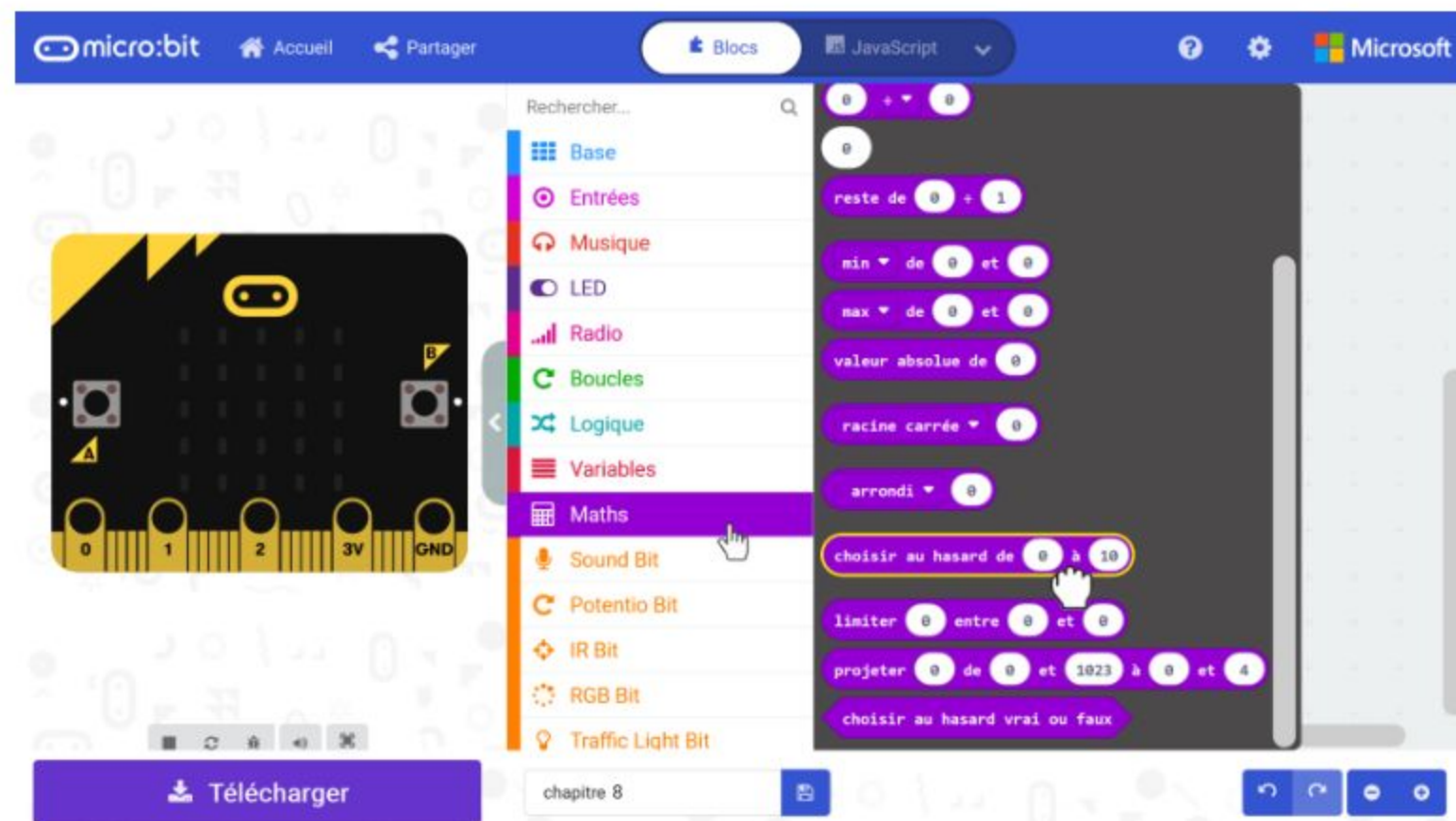


Étape 3 Clique sur la catégorie [**Base**] et sélectionne le bloc [**pause _ (ms)**]. Intercalle-le entre les blocs [**démarrer moteur _ à la vitesse _**] et [**arrêter moteur _**].

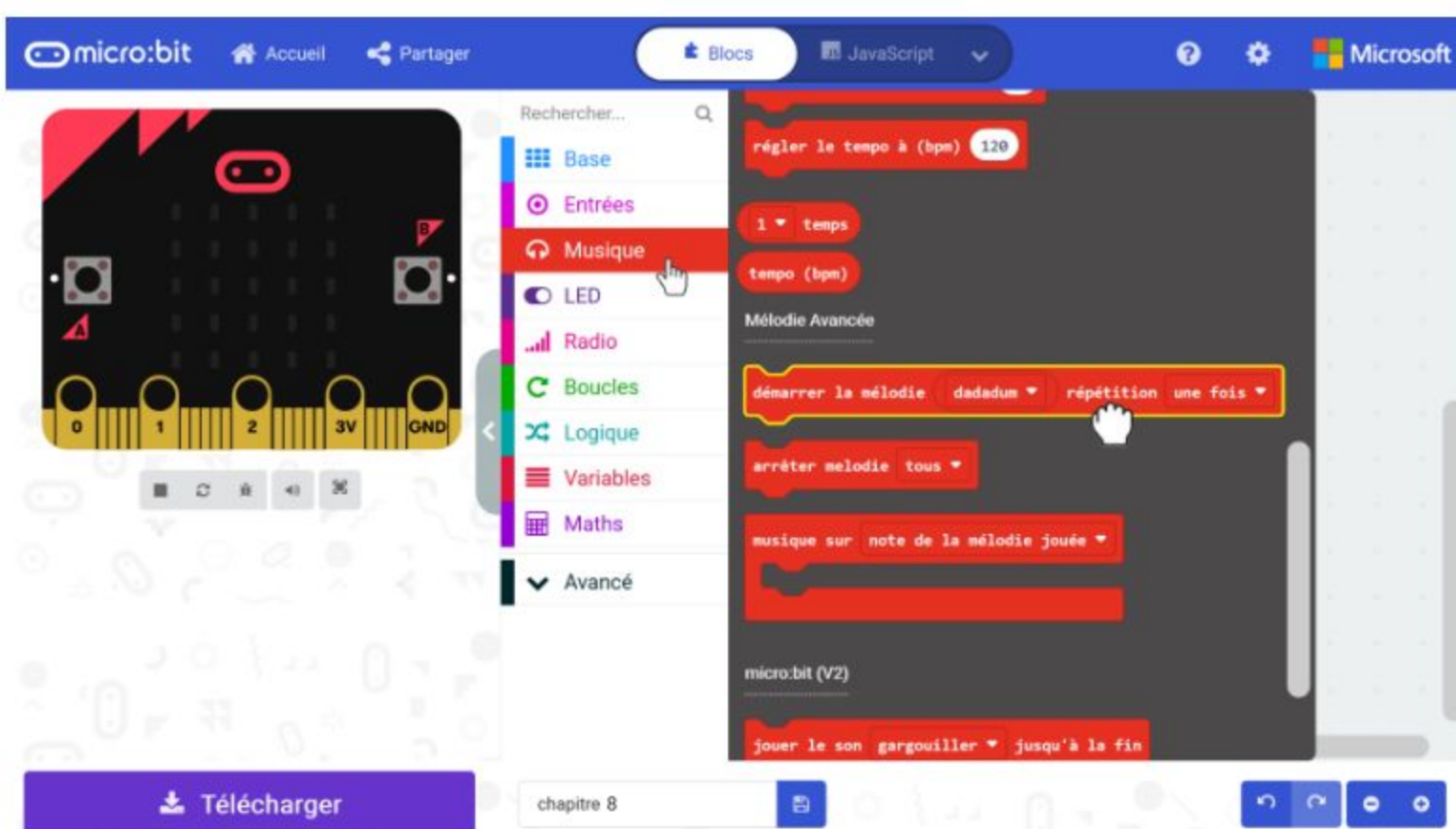




Étape 4 Clique sur la catégorie [**Maths**] et sélectionne le bloc [**choisir au hasard de _ à _**]. Insère-le dans le bloc [**pause (ms)**] et change les bornes en **200** et **1 000** respectivement.



Étape 5 Clique sur la catégorie [**Musique**] et sélectionne le bloc [**démarrer la mélodie _ répétition _**]. Change la mélodie en "**ba ding**" (ou choisis la mélodie que tu veux).

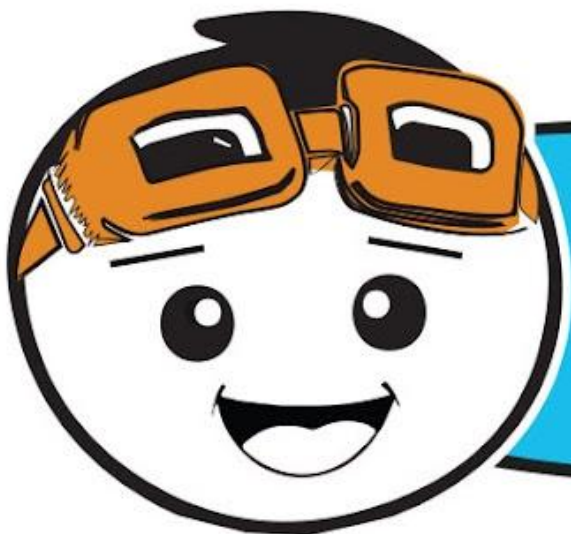
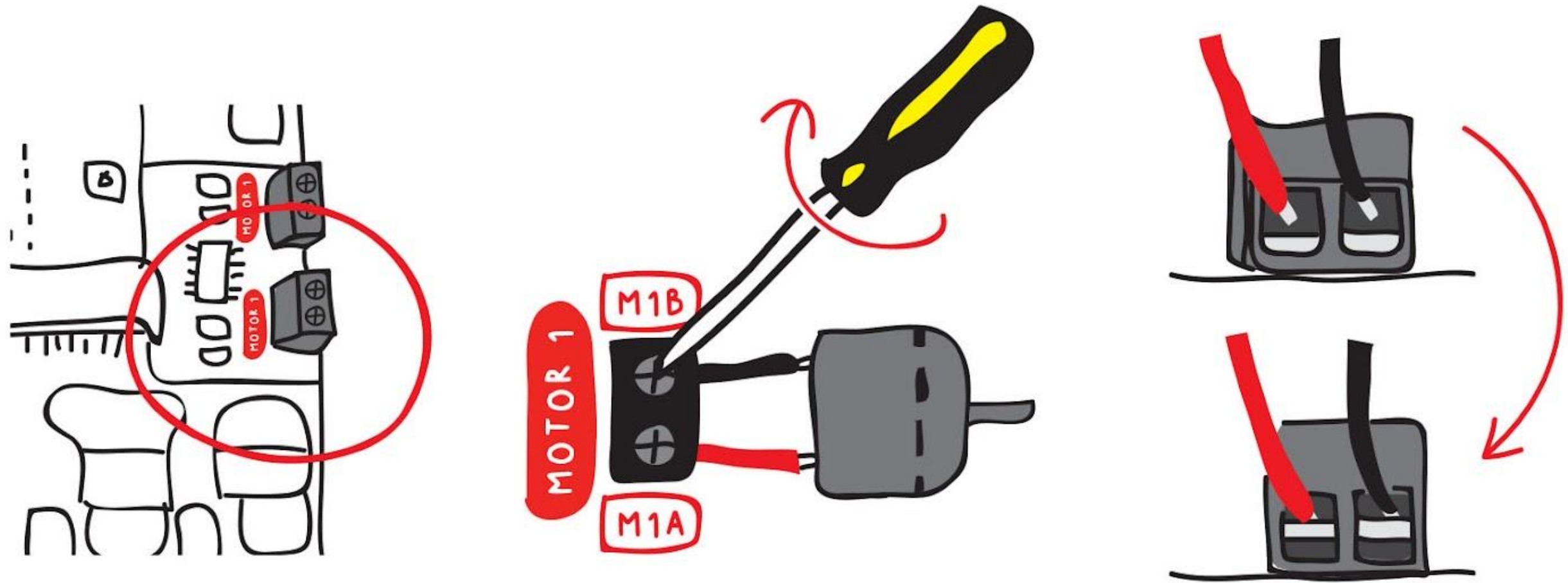


Étape 6 Flashe le code complet sur EDU:BIT.

Nous pouvons utiliser ce code pour n'importe quel jeu utilisant une roue aléatoire (ou roulette). Simplement, le moteur démarre lorsqu'il est déclenché, puis s'arrête après une courte durée déterminée au hasard.

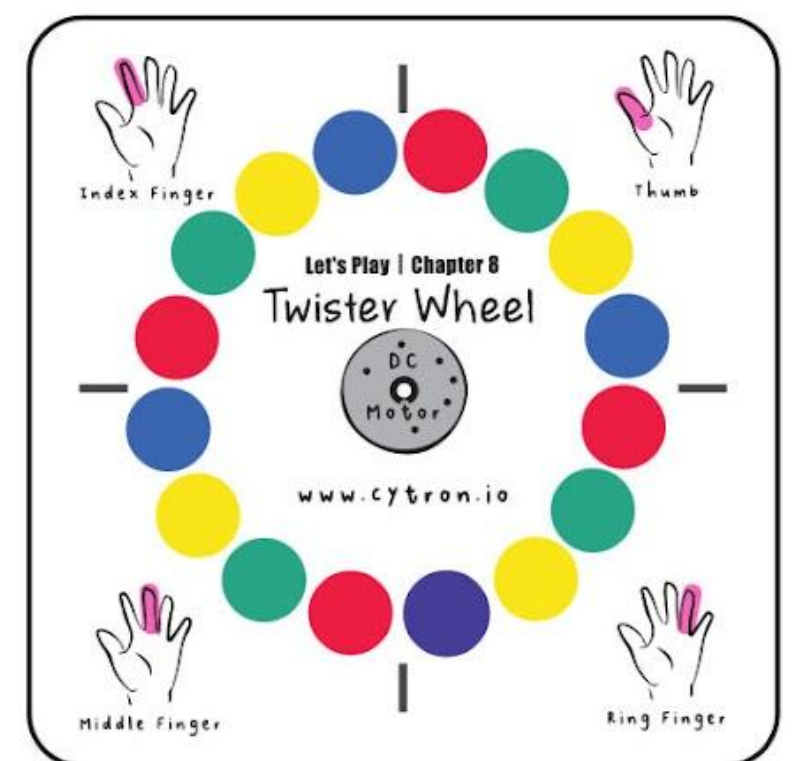
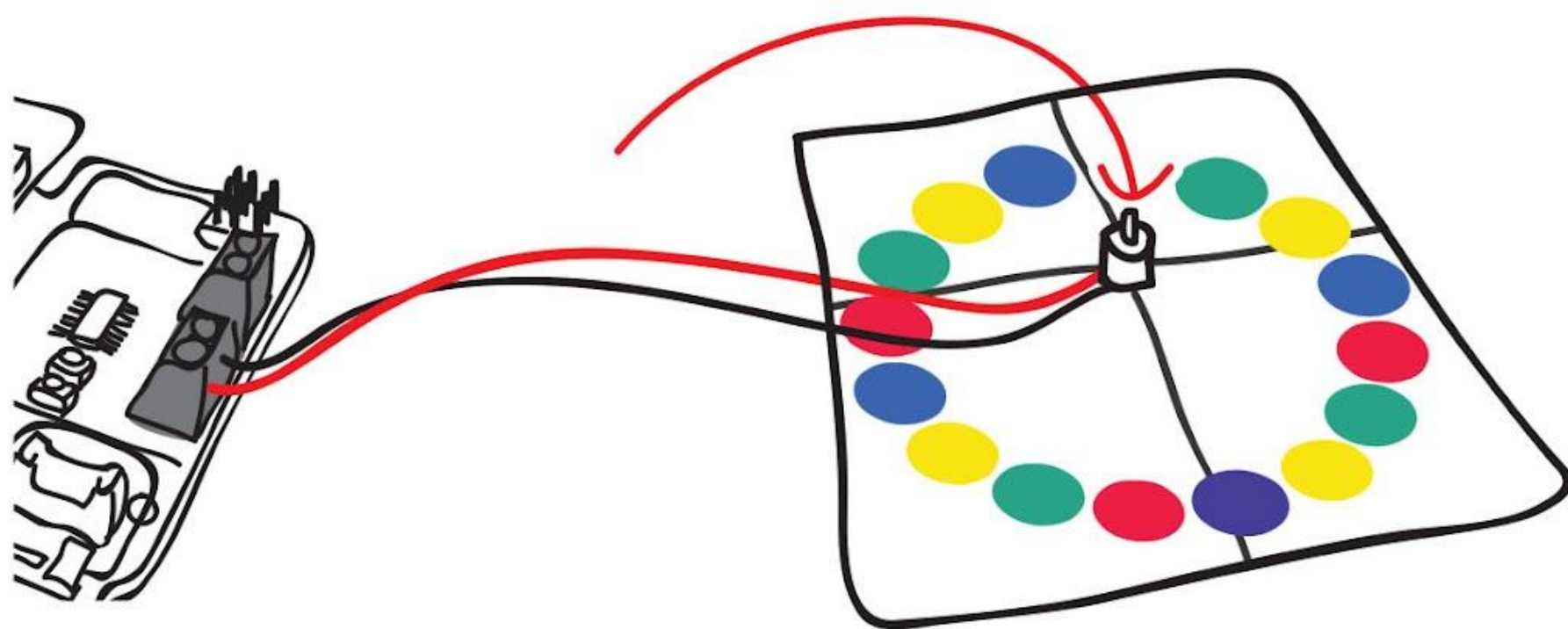


Étape 7 Connecte le moteur DC sur le terminal MOTOR 1 : (i) insère le fil dénudé et (ii) serre la vis à l'aide du tournevis fourni pour sécuriser et bien maintenir la connexion en place.

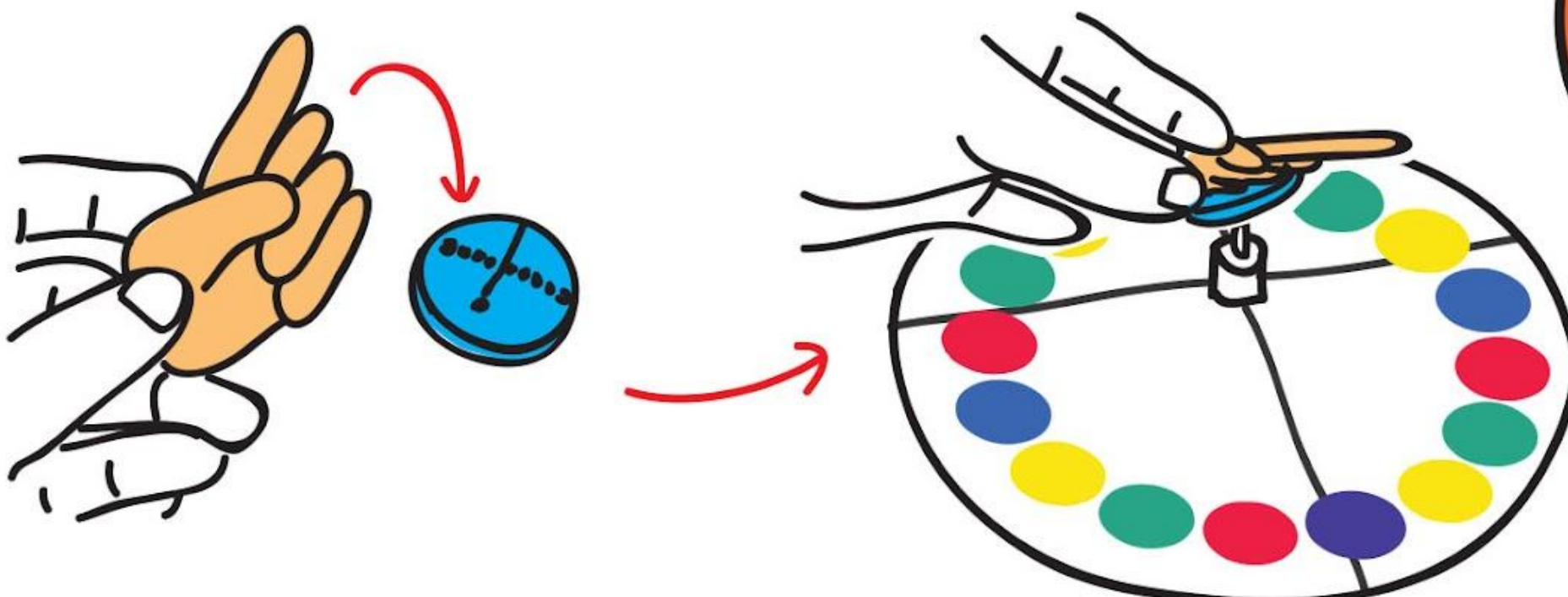


Appuie sur le bouton jaune (bouton A) pour tester. Si le moteur ne tourne pas, vérifie le branchement sur le terminal et vérifie qu'EDU:BIT est bien allumé.

Étape 8 Fixe le moteur DC au centre du plateau "Twister Wheel" (comme indiqué) à l'aide d'un ruban adhésif double face ou de la colle forte.



Étape 9 Dépunche le pointeur et fixe-le sur le disque en plastique avec de l'adhésif. Enfin, fixe le disque sur l'arbre du moteur.



Le plateau "Twister Wheel", le pointeur et le plateau de jeu sont fournis dans la boîte.



Jouons

Allons Faire un Tour !

Préparation du jeu :

- Pose le plateau de jeu sur une table. Deux joueurs s'asseyent face à face. S'il y a plus de joueurs (4 maximum), ils se placent sur les côtés.
- L'arbitre s'assied près d'EDU:BIT pour accéder facilement à la roue.



COMMENT JOUER :

Dans ce jeu, les joueurs essaient de placer leurs doigts sur les disques colorés du plateau de jeu indiqués par l'arbitre.

Le rôle de l'arbitre est d'appuyer sur le bouton jaune (ou bouton A) pour lancer la roue et annonce quel doigt et quelle couleur sont indiqués par le pointeur. Par exemple : index, rouge.

Lorsque c'est ton tour, tu dois écouter les instructions données par l'arbitre et placer le doigt désigné sur un disque de la couleur demandée. Si le doigt désigné est déjà positionné sur un disque de la couleur demandée, tu dois le déplacer vers un autre disque de la même couleur.

Si tu échoues en ne réussissant pas un mouvement, tu es éliminé du jeu.

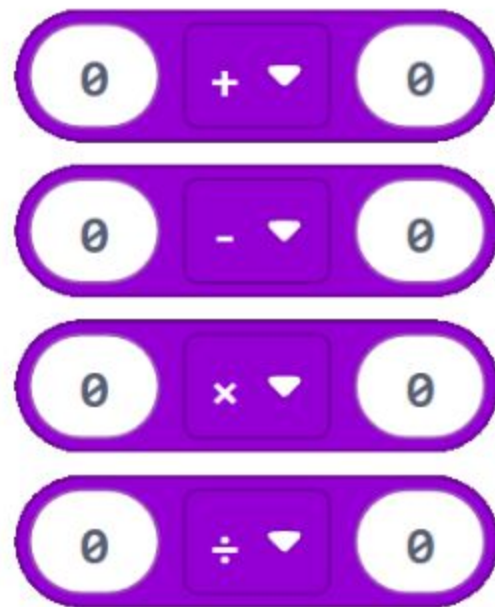
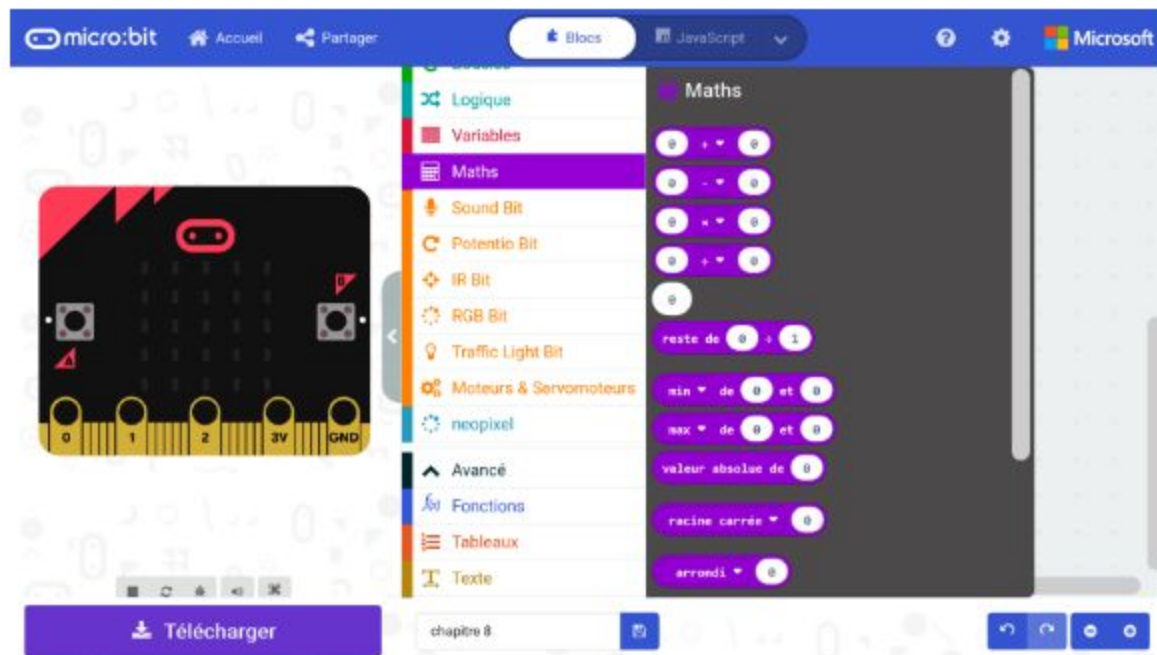
Le dernier joueur encore en jeu est le GAGNANT !



EXPLORE D'AUTRES BLOCS

Tu peux utiliser des blocs de la catégorie [Maths] pour faire des opérations avec tes variables.

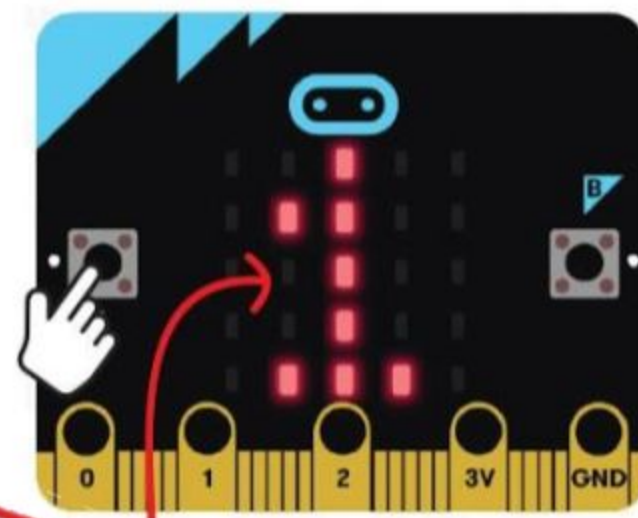
#1 Utilise les blocs suivants pour additionner, soustraire, multiplier et diviser.



#2 Tu peux utiliser le bloc [reste de $_ \div _$] pour trouver le reste dans la division euclidienne d'un entier par un autre.

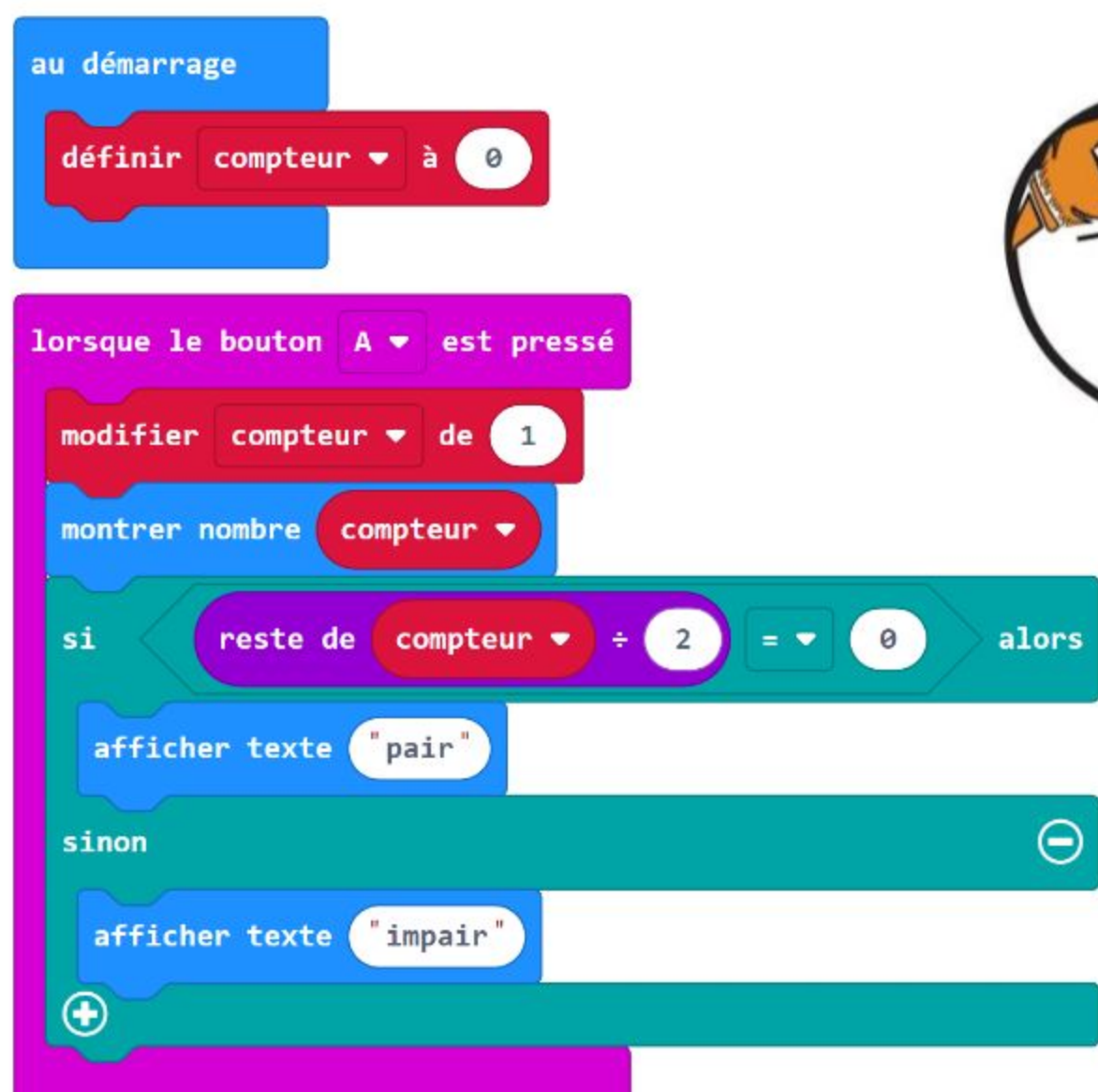


$$\begin{array}{r} 13 \\ - 12 \\ \hline 1 \end{array}$$



reste dans la division euclidienne de 13 par 2.

#3 Tu peux utiliser le bloc [reste de $_ \div _$] pour déterminer si un nombre entier est pair ou impair. On divise un nombre entier par 2 (division euclidienne). Si le reste est 1, alors le nombre est impair. Si le reste est 0, alors le nombre est pair.



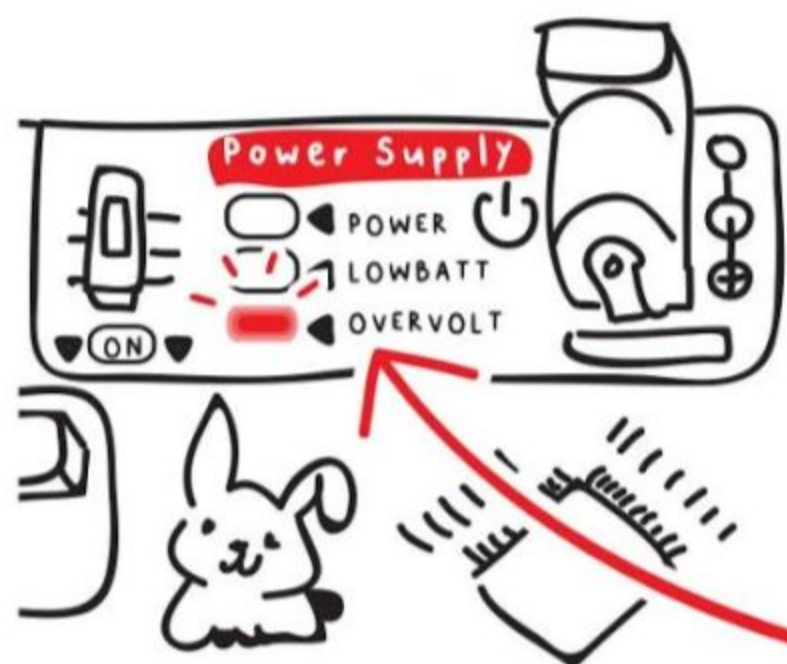
As-tu remarqué qu'EDU:BIT affiche "impair" lorsque tu appuies sur le bouton A pour la première fois (1 = nombre impair) et "pair" lorsque tu appuies une deuxième fois (2 = nombre pair) ? Qu'affichera-t-il lorsque tu appuieras sur le bouton A pour la 99^{ème} fois ?

FAIT AMUSANT !



Un moteur à courant continu, également appelé **moteur DC**, est un dispositif électrique rotatif qui convertit l'énergie électrique reçue en énergie mécanique.

Nous devons appliquer une tension continue pour faire tourner un moteur DC. On peut contrôler la vitesse de rotation en ajustant la tension délivrée. Plus la tension délivrée est grande, pour le moteur tourne vite. La tension recommandée pour le moteur DC du kit EDU:BIT est de 3,6 V à 6 V.



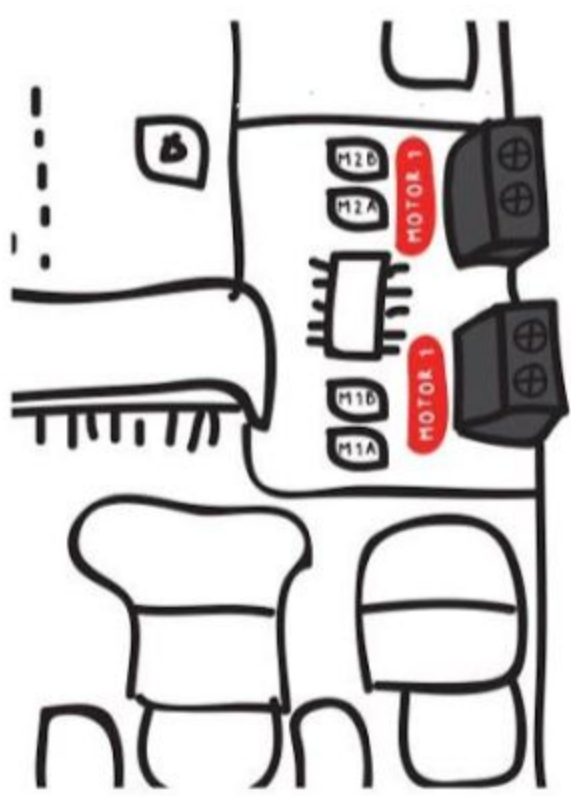
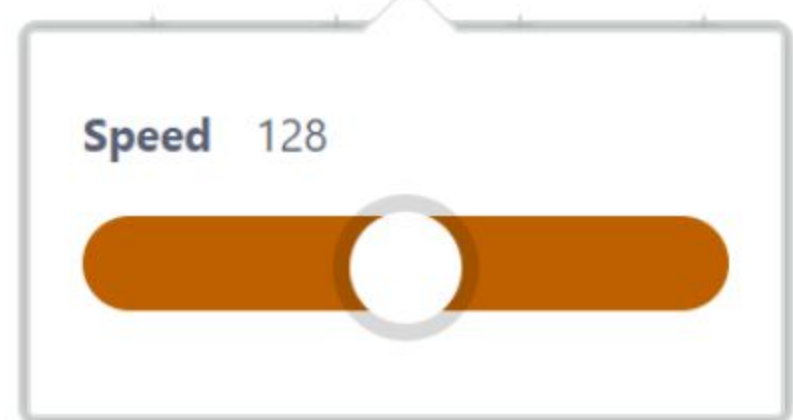
ATTENTION !

Appliquer une tension trop élevée (supérieure aux recommandations) réduira la durée de vie du moteur sur le long terme.



Indicateur de surtension

Tu peux facilement contrôler la direction et la vitesse du moteur DC avec le bloc de programmation suivant.



Direction de rotation.

Il y a deux terminaux moteur DC sur la carte EDU:BIT. Sélectionne le bon terminal.

Valeur relative à la plage de valeurs de 0 à 255. Plus la valeur est grande, plus le moteur tourne vite.

Sais-tu qu'EDU:BIT dispose d'un circuit de test de moteur DC intégré ? Appuie sur les boutons blancs (nommés M1A, M1B, M2A et M2B) pour vérifier que les connexions sont sécurisées et que le moteur fonctionne bien.



APPLICATION DÉFI

Programme EDU:BIT pour qu'elle serve d'interrupteur sonore pour un ventilateur dont la vitesse est contrôlée par Potentio Bit.

au démarrage

Afficher l'icône cœur (ou l'icône de ton choix).
Définir la variable mode à 0.

lorsque volume > _
(valeur seuil)

Incrémenter la variable mode de 1.

toujours

Définir la variable "vitesse" par mappage de la valeur du potentiomètre comprise entre 0 et 1 023 en une valeur comprise entre 0 et 255.

Vérifier constamment mode

- Si mode est un nombre pair, alors arrêter le moteur M1.
- SINON Si mode est un nombre pair, alors démarrer le moteur M1 à une vitesse définie par la variable "vitesse" (mappée à partir de la valeur du potentiomètre lue).



Quelques conseils pour toi...

#1: Tu dois déterminer le volume de déclenchement (ie : la valeur seuil) pour démarrer ou arrêter le moteur.

#2: Tu as besoin de créer deux variables : mode et vitesse.

#3: Attache l'hélice sur l'arbre du moteur puis lance le programme. Si tu ne sens pas le vent souffler lorsque le moteur tourne, tu dois changer le sens de rotation (ie : la direction) dans ton code.

CHAPITRE 9

Tir au But... Goal!

Servomoteur

Readyy..

Get set..

GO!!!!!!



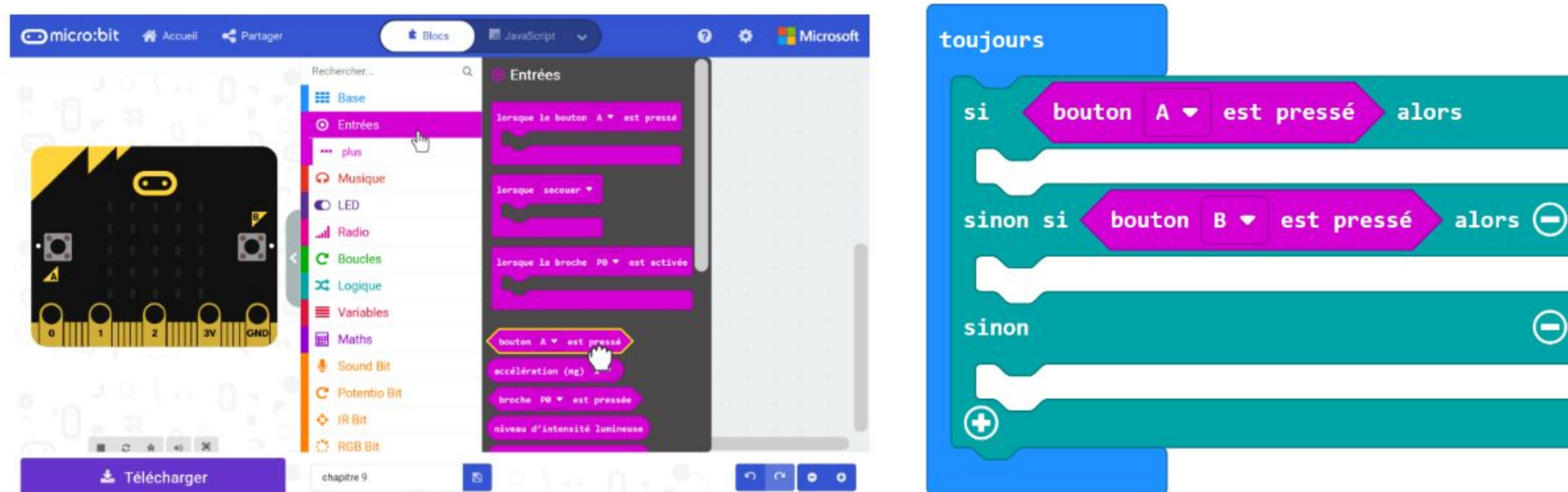
link.cytron.io/edubit-chapter-9

CODONS !

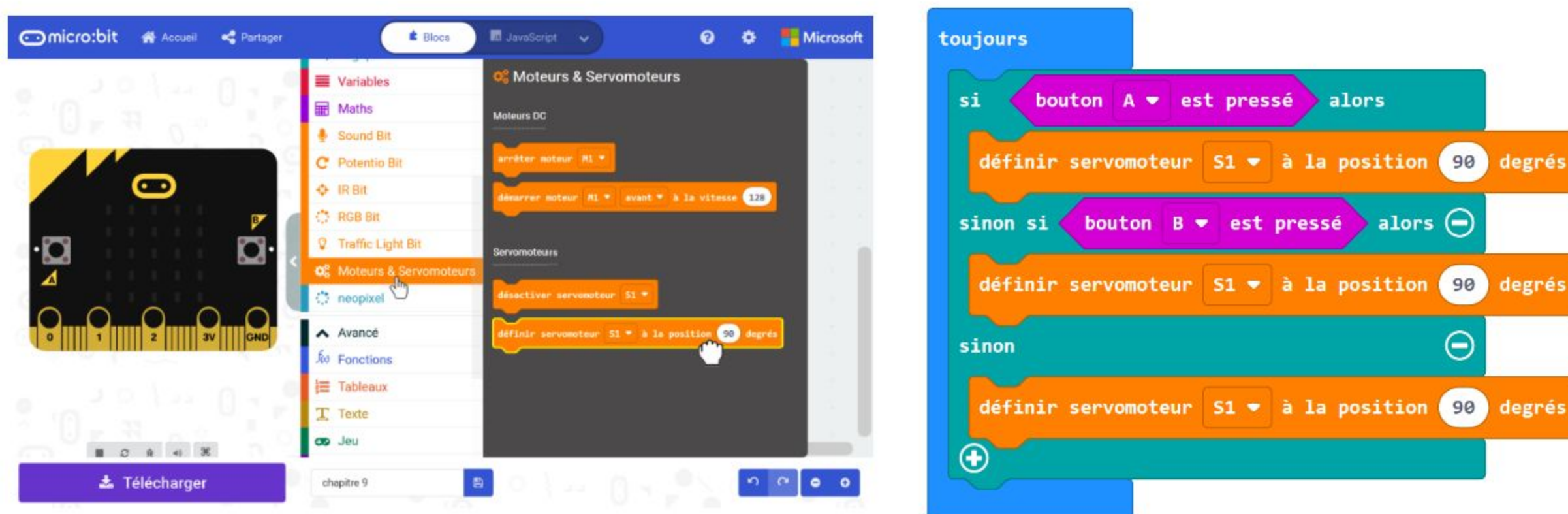
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT (voir page 40). Clique sur la catégorie [**Logique**] et sélectionne le bloc [**si-alors-sinon**]. Imbrique-le dans le bloc [**toujours**]. Clique sur l'icône + pour ajouter un test sinon-si.



Étape 2 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**bouton _ est pressé**]. Duplique-le et insère-les dans les conditions du bloc [**si-alors-sinon**]. Change le bouton du second bloc en "B".



Étape 3 Clique sur la catégorie [**Moteurs & Servomoteurs**] et sélectionne le bloc [**définir servomoteur _ à la position _ degrés**]. Duplique-le et imbrique-les dans chaque emplacement du bloc [**si-alors-sinon**].



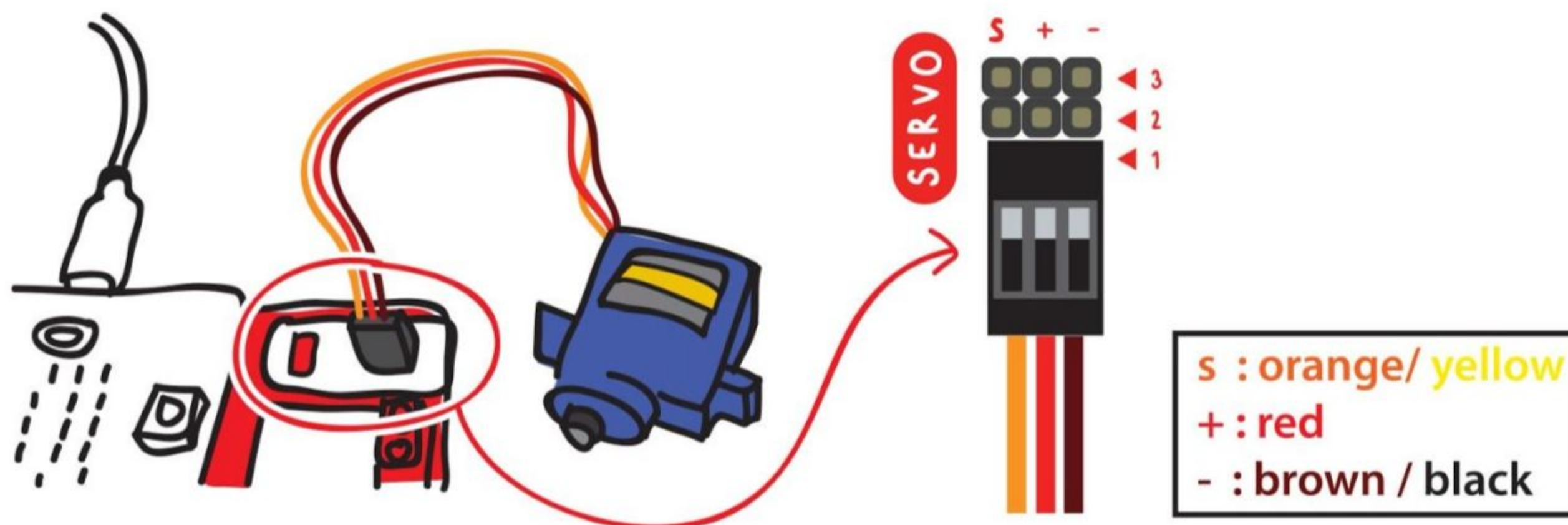


Étape 4 Change les valeurs des deux premiers blocs en **30** et **150** respectivement. Flashe le code sur EDU:BIT.

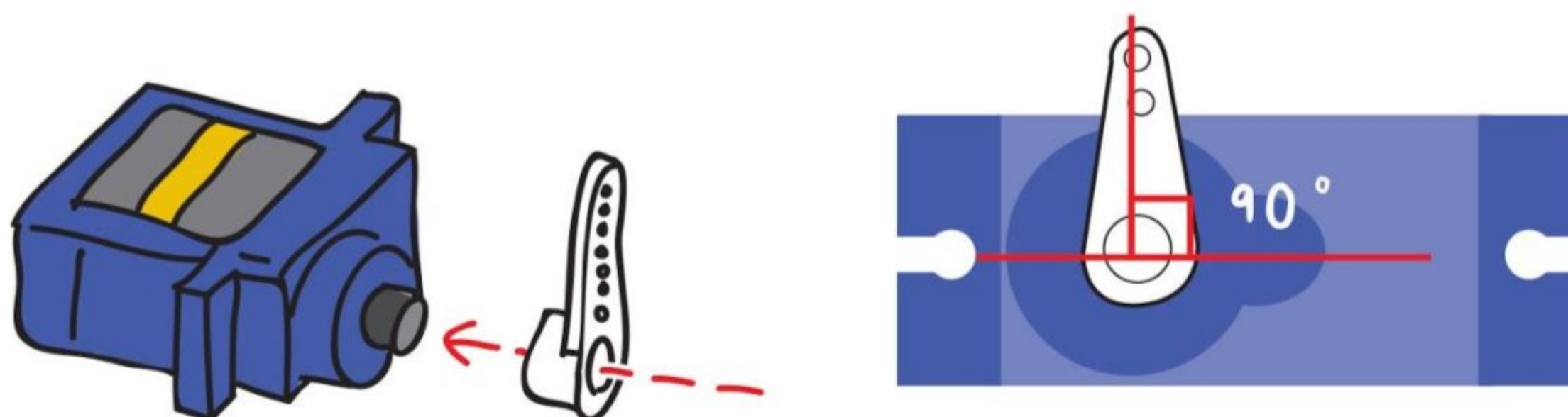
```

toujours
si bouton A ▼ est pressé alors
définir servomoteur S1 ▼ à la position 30 degrés
sinon si bouton B ▼ est pressé alors
définir servomoteur S1 ▼ à la position 150 degrés
sinon
définir servomoteur S1 ▼ à la position 90 degrés
    
```

Étape 5 Branche le servomoteur sur le Servo Port 1 de la carte EDU:BIT comme montré ci-dessous.

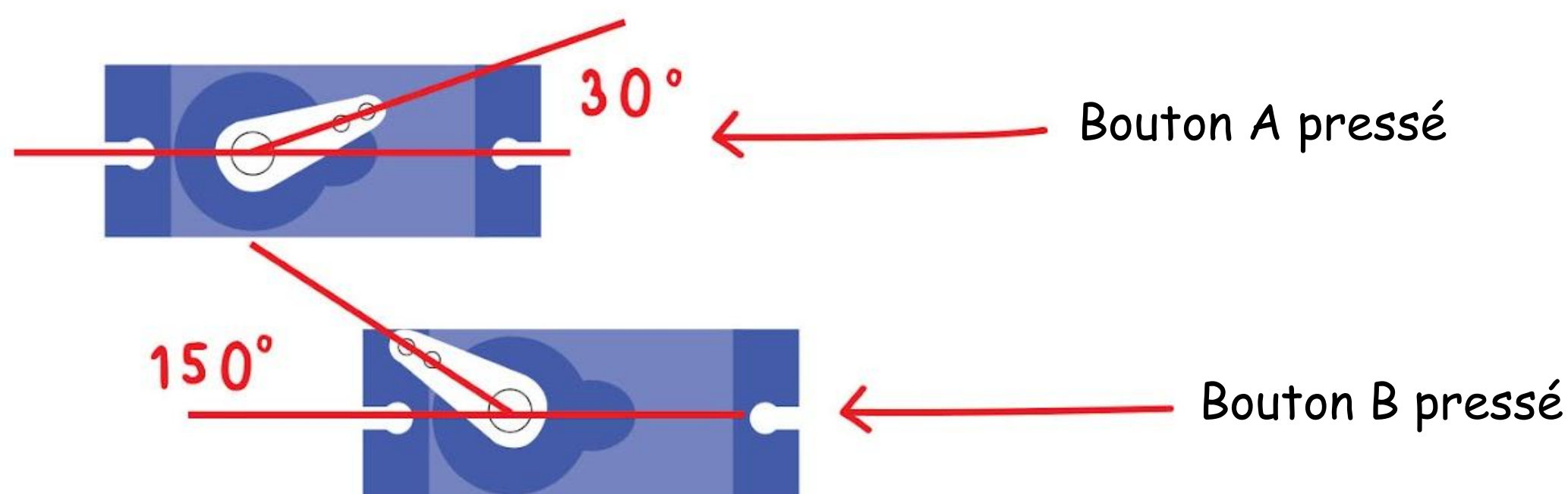


Étape 6 Allume EDU:BIT et clippe le bras de fixation sur l'arbre du servomoteur à 90 degrés, comme montré ci-dessous.

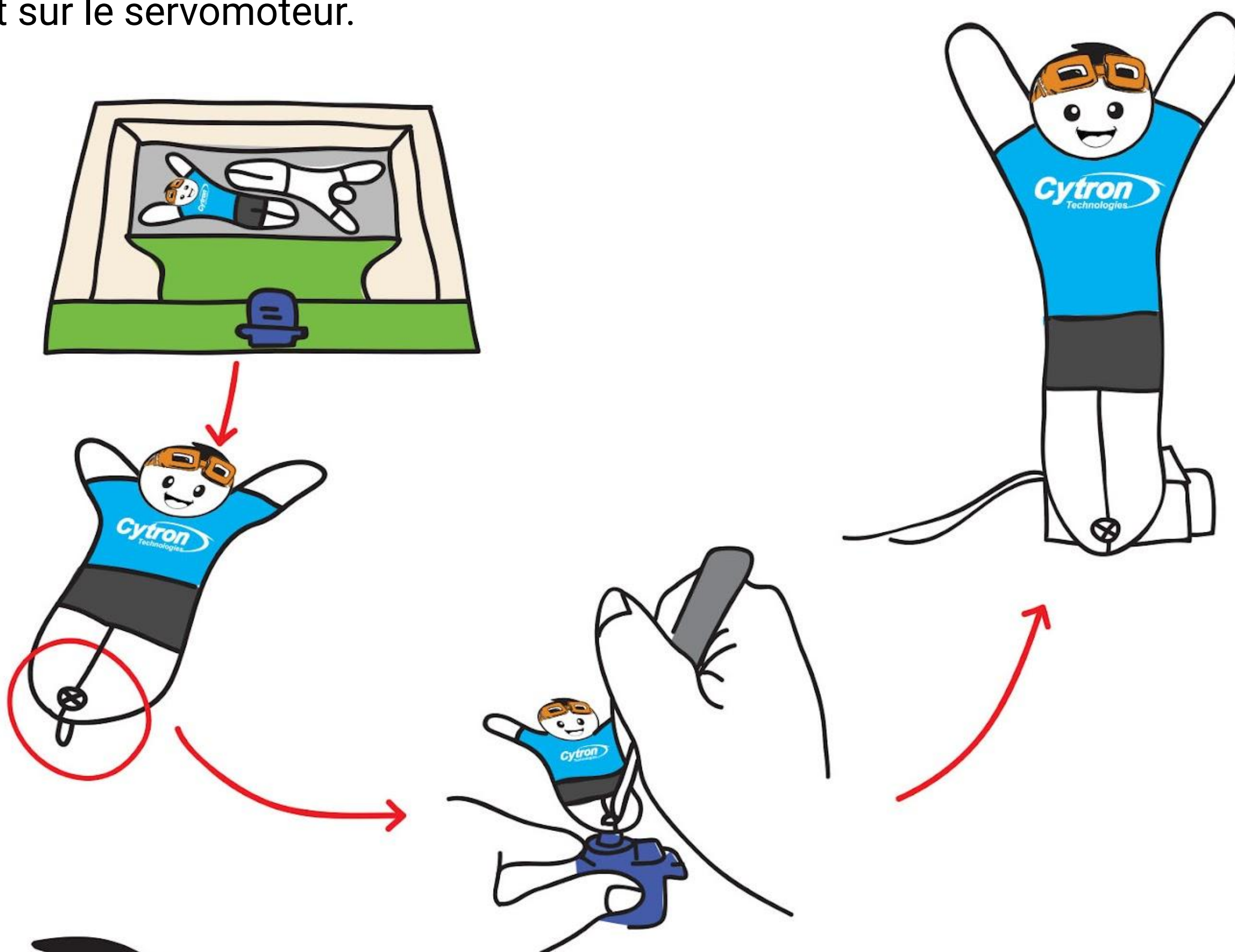


CHAPITRE 9 : Tir au But... Goal !

Étape 7 Appuie sur le bouton A, puis sur le bouton B, pour tester.



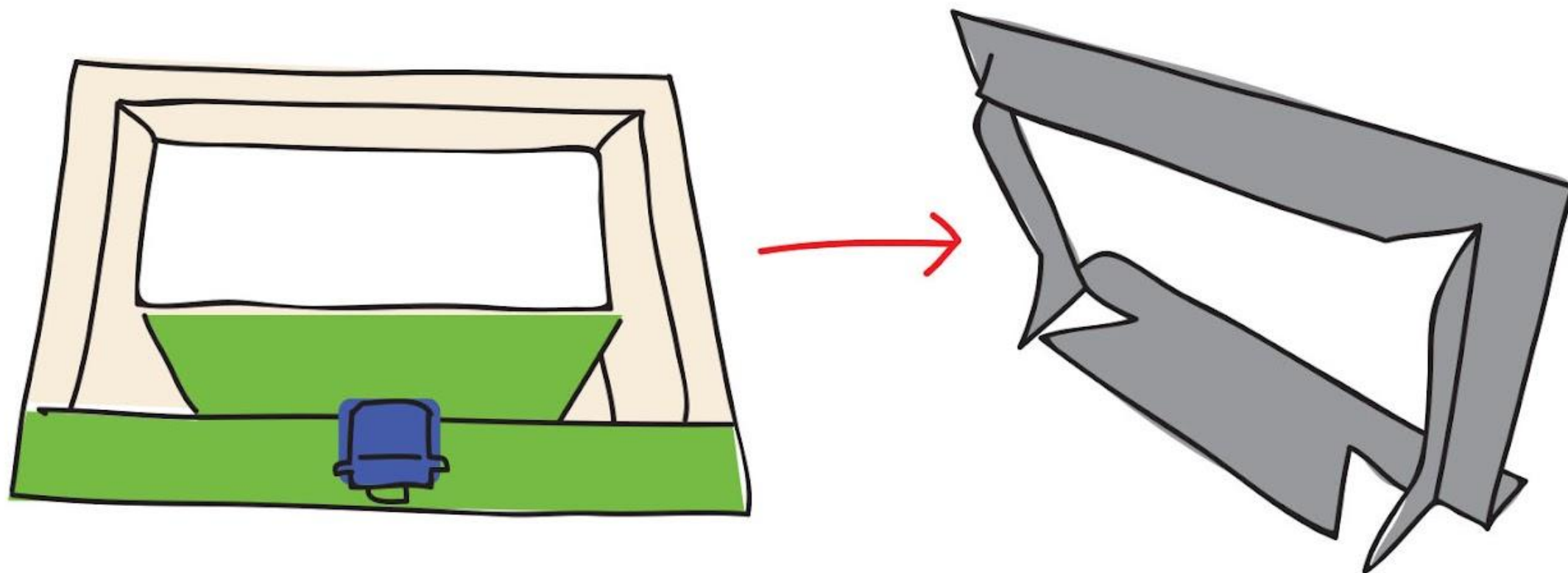
Étape 8 Dépuncher le gardien de but de la plaquette fournie dans le kit. Utilise une vis et le tournevis fournis pour fixer le gardien de but sur le bras du servomoteur. Utilise du ruban adhésif double face ou de la colle forte pour consolider la fixation du gardien de but sur le servomoteur.



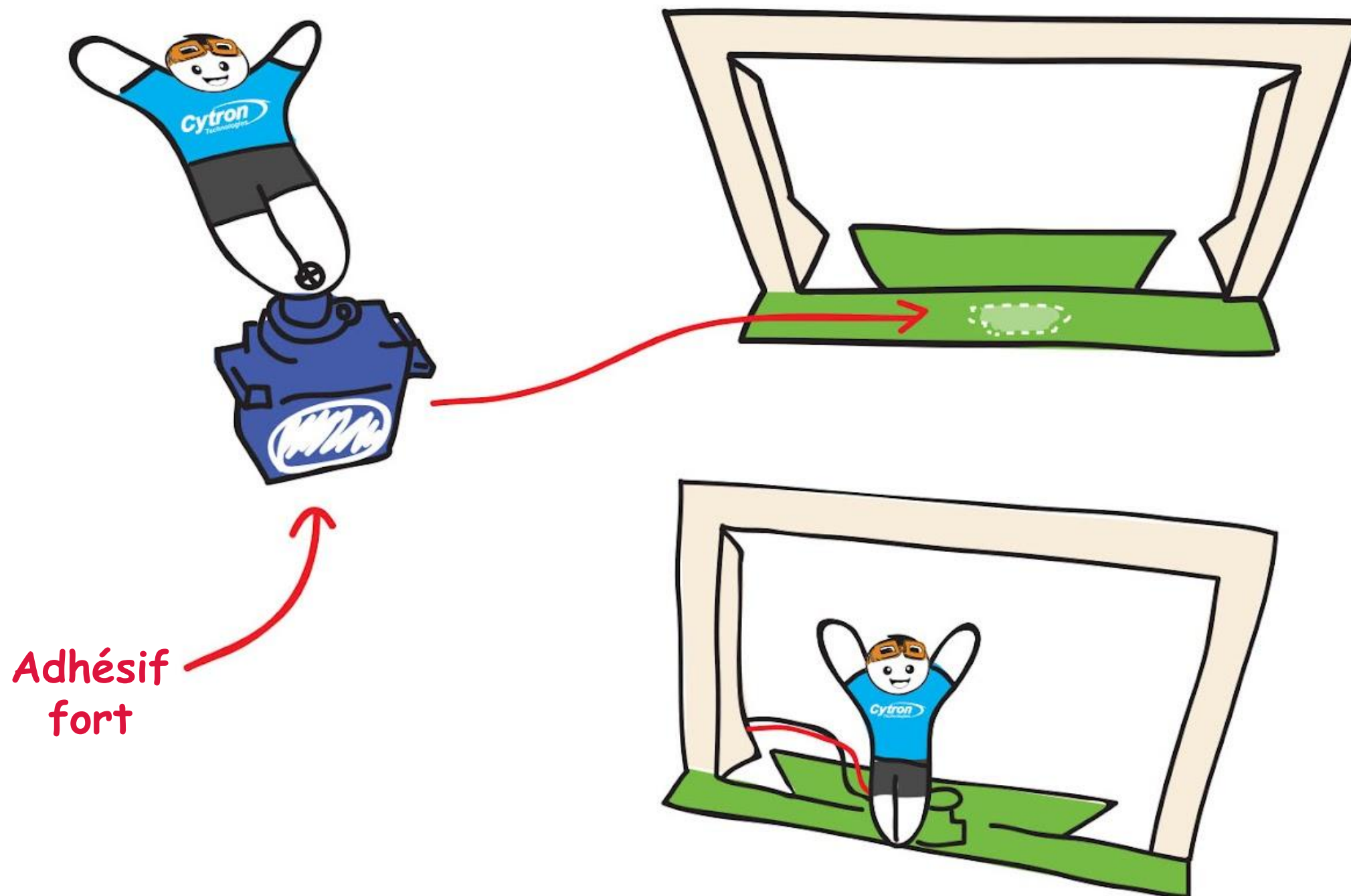
Si tu veux, tu peux dépuncher l'autre partie, blanche, et y dessiner un gardien de but portant un maillot aux couleurs de ton choix.



Étape 9 Dépunche le but et dresse-le comme montré ci-dessous.

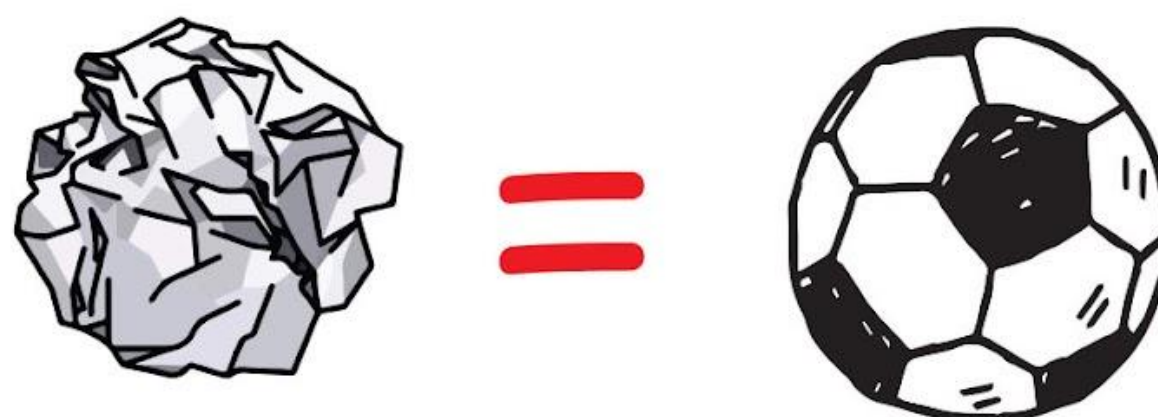


Étape 10 Utilise du ruban adhésif double face ou de la colle forte pour fixer fermement le servomoteur à la place indiquée.



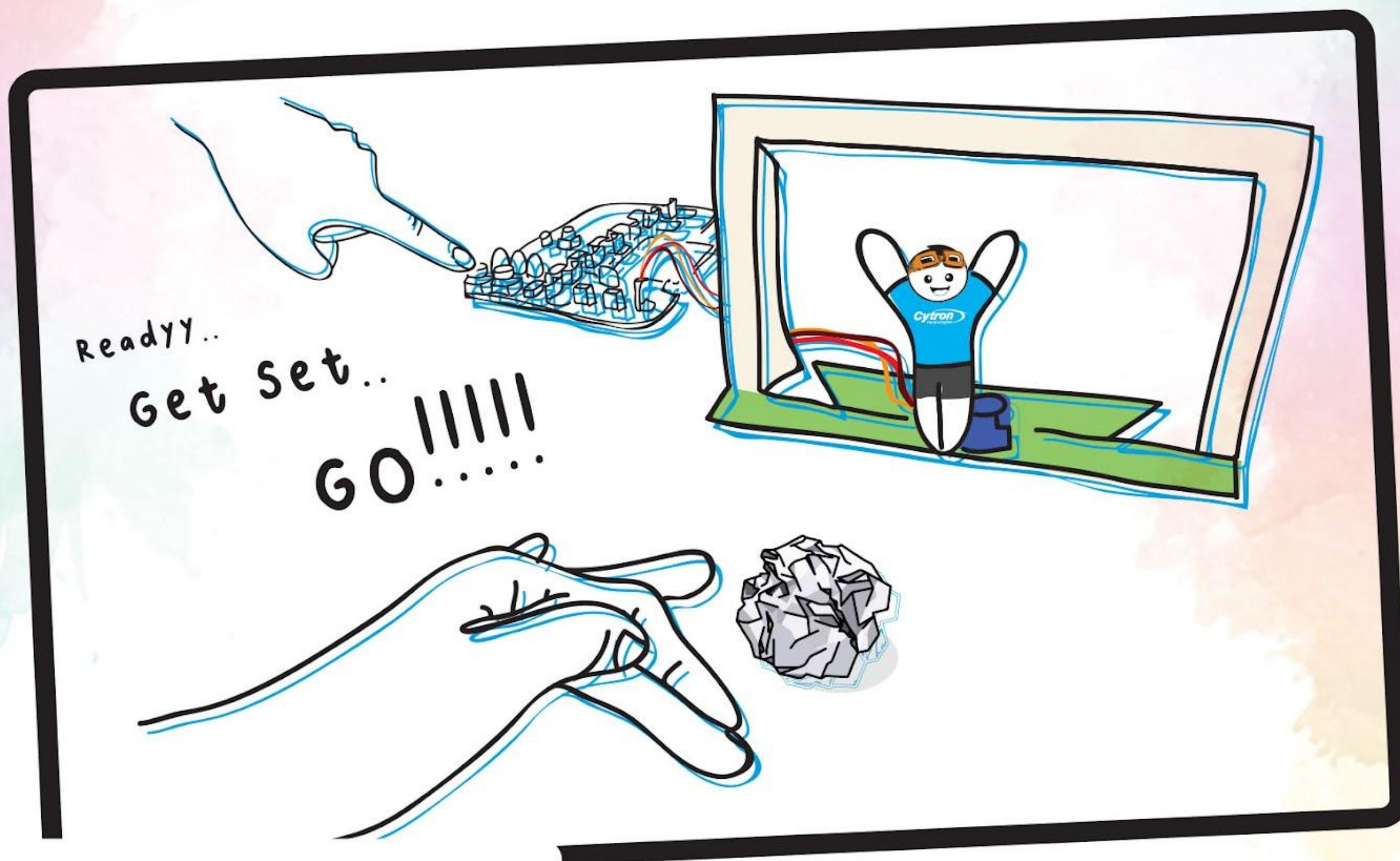
Adhésif fort

Étape 11 Fabrique une boulette de papier comme substitut de ballon de football et ainsi, tout est en place pour se faire une partie de tirs au but. Es-tu prêt ?



Jouons

Tir au But... Goal !!!



COMMENT JOUER :

Placer le but et définir le point de pénalty (à 1 mètre du gardien, réduire la distance pour des joueurs plus jeunes).

Les joueurs sont le tireur et le gardien de but à tour de rôle.

Le tireur frappe le ballon à l'aide d'une pichenette en direction du but.

Le gardien de but essaie d'arrêter le ballon en se déplaçant à gauche (bouton jaune ou bouton A) ou à droite (bouton bleu ou bouton B).

Dans une manche, chaque joueur a 5 tentatives. Le joueur qui marque le plus de buts gagne.



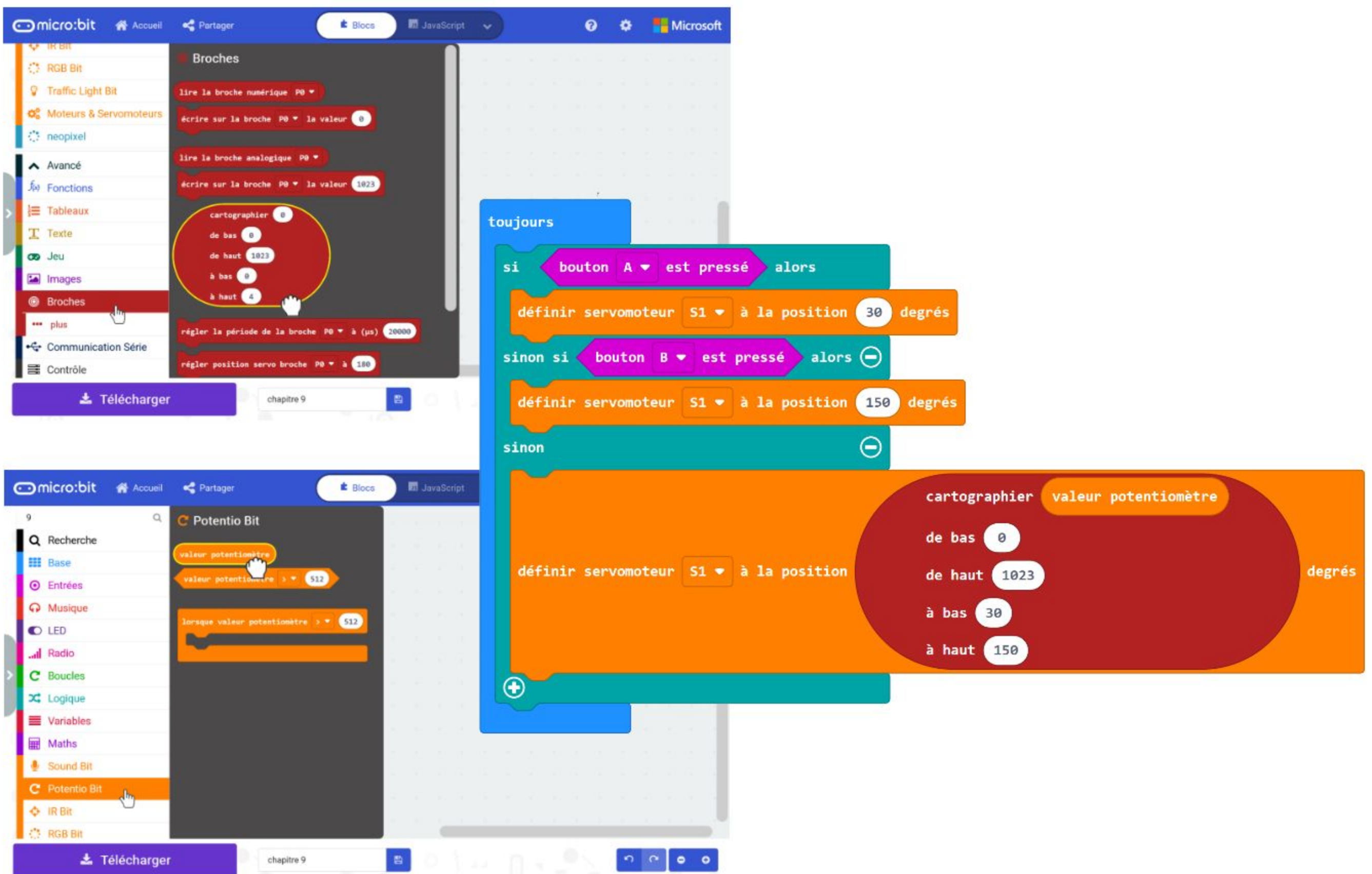
Le savais-tu ? Une séance de tirs au but permet de déterminer l'équipe qui remporte le match en cas d'égalité à la fin du temps réglementaire, et restent à égalité après les prolongations (s'il y en a). Dans une séance de tirs au but, chaque équipe tire 5 fois. Le joueur tire depuis le point de pénalty en direction du but qui est protégé uniquement par le gardien de but de l'équipe adverse. L'équipe qui a marqué le plus de buts est le vainqueur du match.



Dans notre code précédent, nous avons balancé le gardien de but à gauche ou à droite en appuyant sur des boutons. Nous pouvons modifier le code pour contrôler la position du gardien à l'aide du Potentio Bit.

Étape 12 Clique sur [**Avancé**] puis sur la catégorie [**Broches**]. Ajoute le bloc [**cartographeur _ de bas _ de haut _ à bas _ à haut _**] à ton code.

Étape 13 Clique sur la catégorie [**Potentio Bit**] et sélectionne le bloc [**valeur potentiomètre**]. Insère-le dans le bloc [**cartographeur _ de bas _ de haut _ à bas _ à haut _**] et change les deux dernières valeurs en **30** et **150** respectivement.



Étape 14 Flashe le code sur EDU:BIT. Tu peux contrôler le gardien de but à l'aide du Potentio Bit. Amuse-toi !

Si tu veux t'entraîner à faire des tirs au but, tu peux modifier le code en "mode entraînement", et faire basculer le gardien de but de gauche à droite continuellement. Essaie !



FAIT AMUSANT!

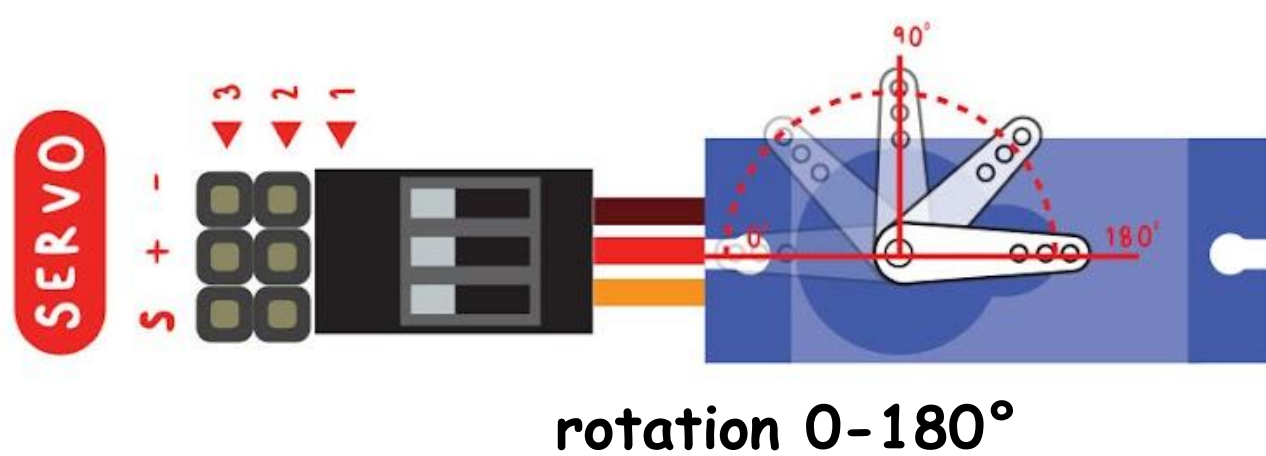
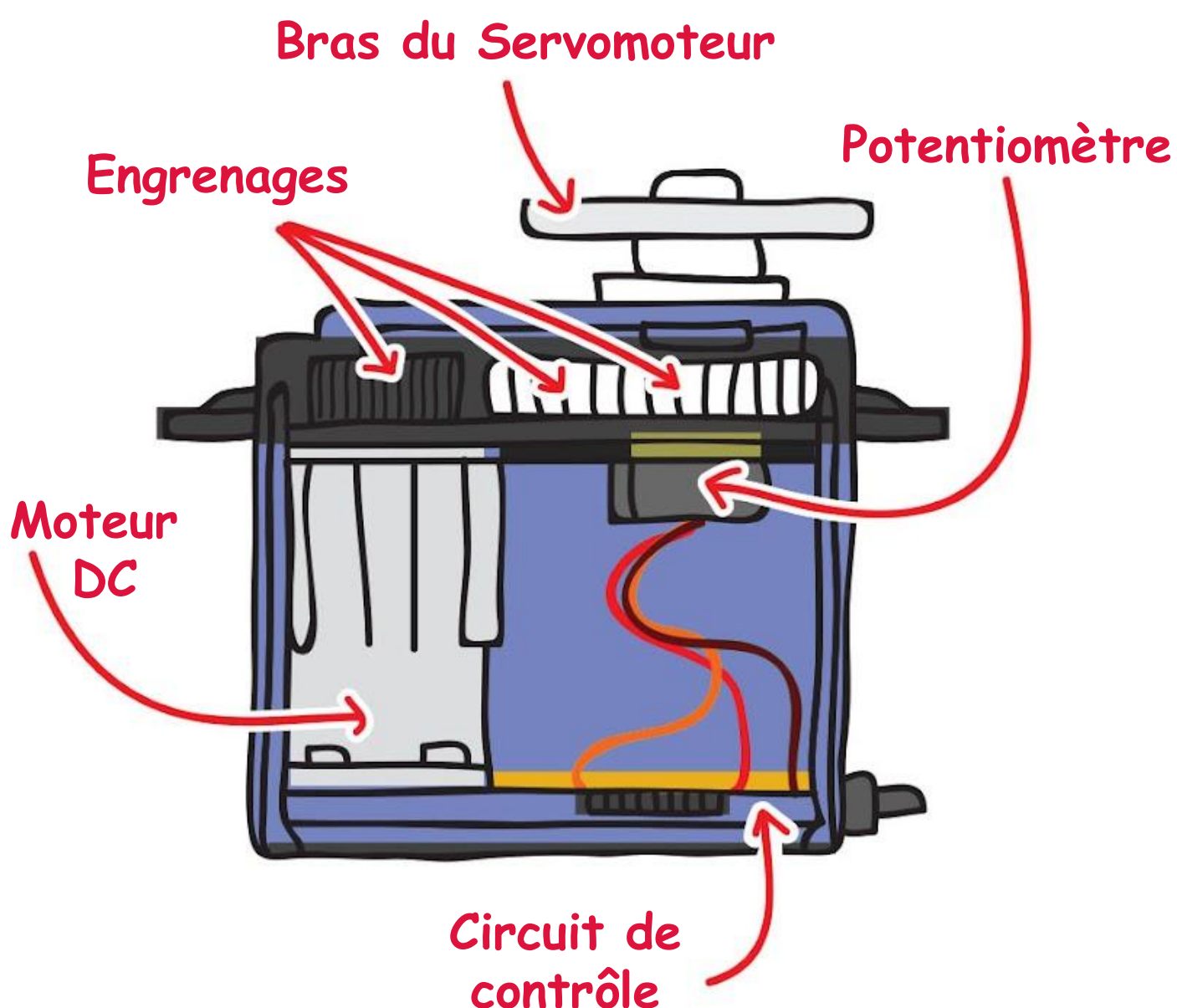


Le **servomoteur** du kit EDU:BIT est aussi connu sous le nom de servo RC (Radio Control). Il est largement utilisé dans les jouets radiocommandés (comme les voitures) et les petits robots pour contrôler leurs mouvements.

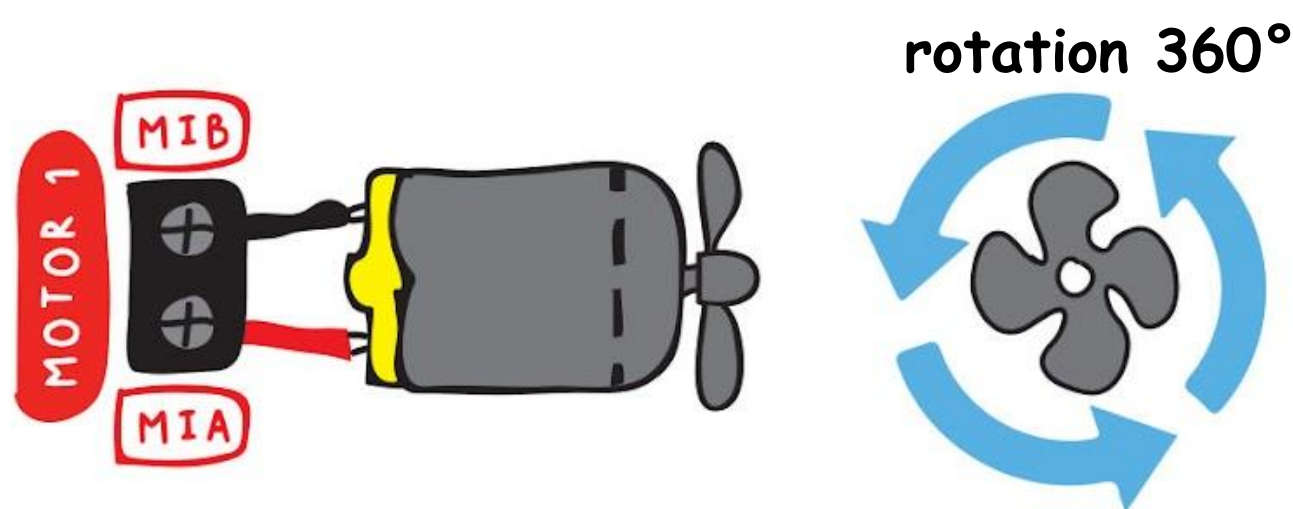
Un servomoteur utilise une connexion 3 fils pour l'alimentation (+), la masse (-) et le signal de contrôle(s). Il est généralement constitué d'un moteur DC, d'engrenages, d'un potentiomètre (capteur de position) et d'un circuit de contrôle.

Le contrôleur intégré traduit les commandes sous forme d'impulsions pour le positionner à l'angle demandé. Le servomoteur va constamment tourner puis rester à la position correspondant aux impulsions reçues.

Contrairement aux moteurs DC qui tournent continuellement, on peut contrôler la rotation d'un servomoteur avec l'angle que l'on souhaite, compris entre 0 et 180°.



Servomoteur VS Moteur DC



APPLICATION DÉFI

Programme EDU:BIT pour qu'il serve de métronome. Lorsqu'il est allumé, il bascule continuellement le pointeur (fixé sur le bras du servomoteur) de gauche à droite à un tempo constant. Le tempo est défini par le Potentio Bit. Lorsque le bouton jaune (bouton A) est pressé, il affiche le tempo actuel (par exemple : 120 bpm).

Voici quelques conseils pour toi...

- #1: Tu dois créer deux variables : tempo et délai.*
- #2: Un métronome standard donne un tempo de 40 à 200 bpm.*
- #3: Un tempo de 60 bpm (battement par minute) signifie que le pointeur bascule d'un bout à l'autre 60 fois au total en une minute, ie : une fois par seconde.*
- #4: Plus rapide est le tempo, plus court est le délai.*



"Un métronome est un instrument qui produit un bip audible ou un autre son à intervalles de temps réguliers qui peuvent être définis par l'utilisateur, usuellement en battement par minute (bpm). Les musiciens l'utilisent pour travailler le rythme. Les métronomes sont souvent accompagnés d'un effet visuel synchronisé avec le tempo (par exemple : un pendule oscillant ou une lumière qui clignote)."

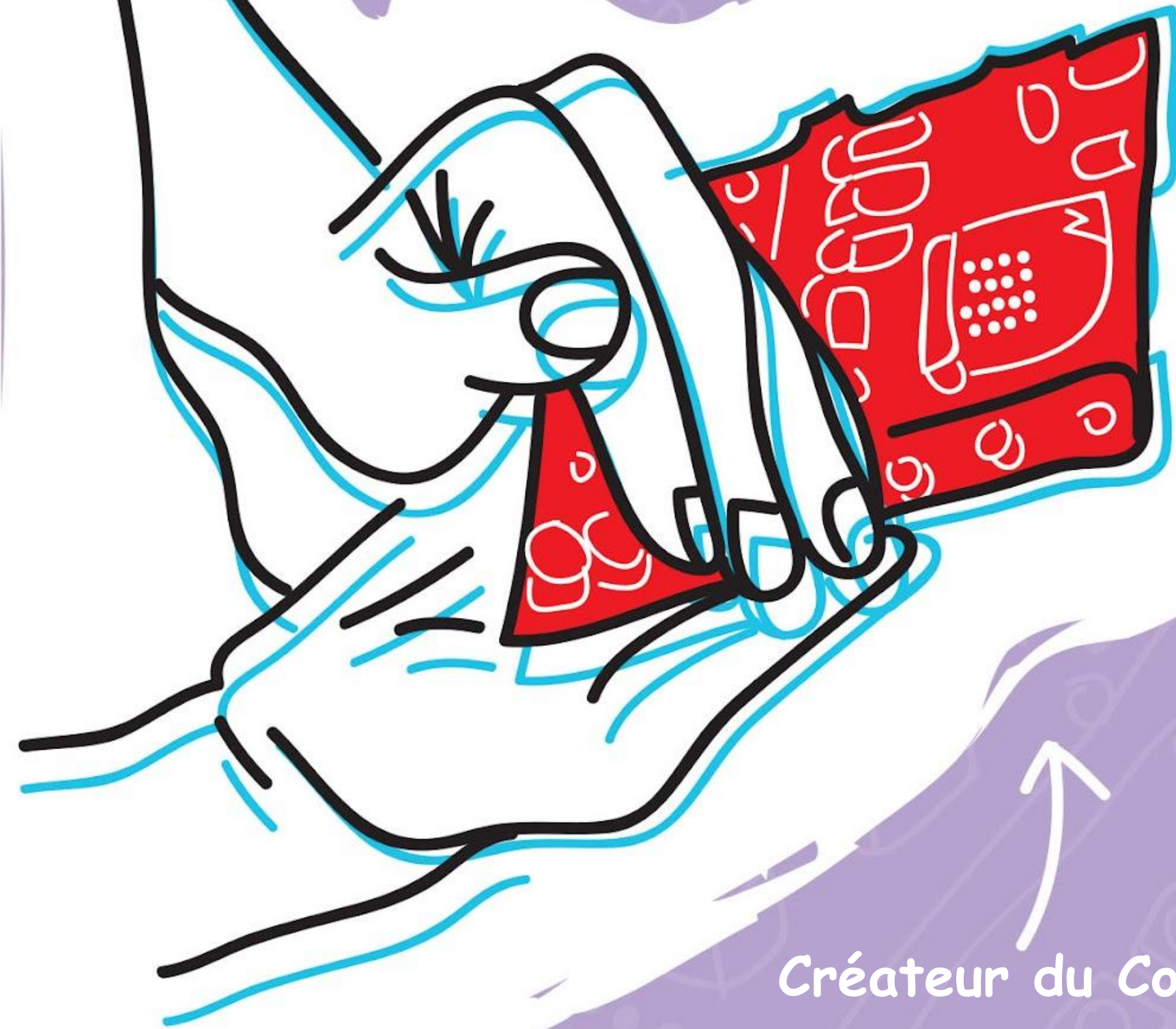
- wikipedia -

CHAPITRE 10

Mastermind, Peux-Tu Déchiffrer le Code ?

RGB Bit

Casseur de Code



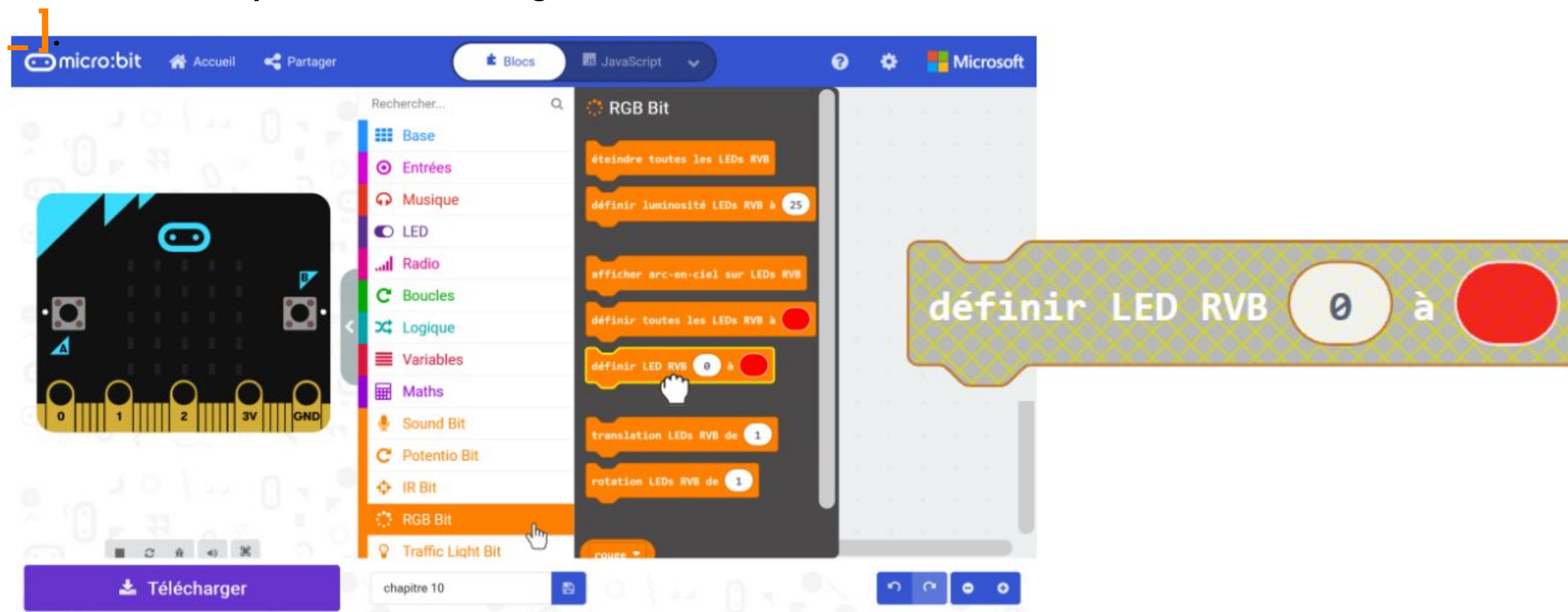
Créateur du Code



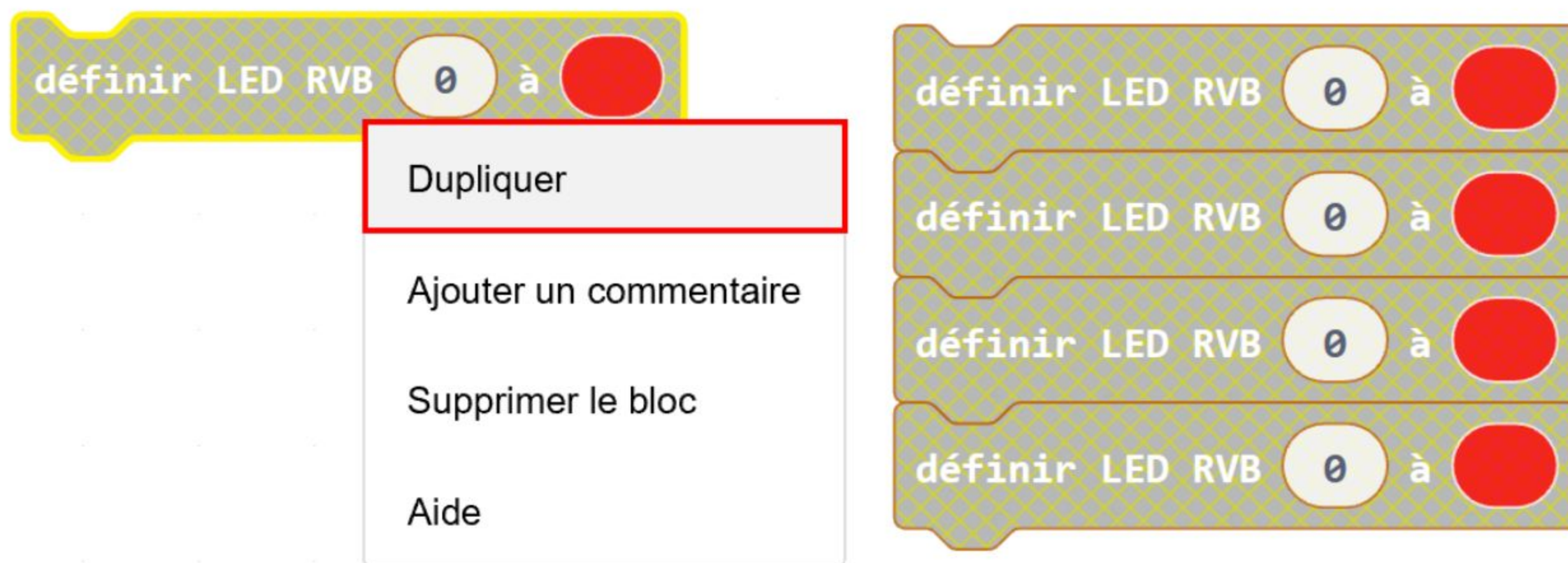


CODONS !

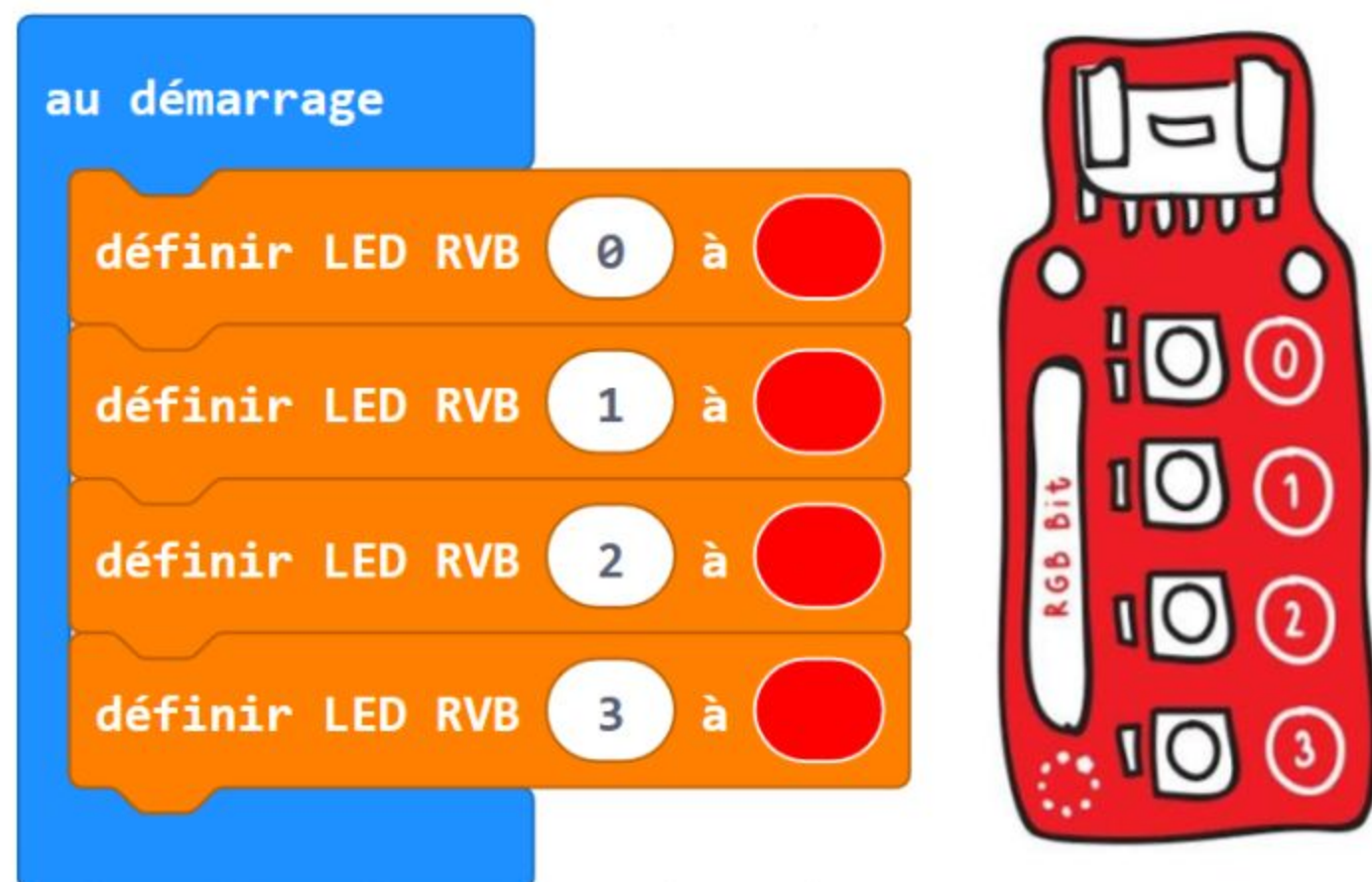
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT. Clique sur la catégorie [**RGB Bit**] et sélectionne le bloc [**définir LED RVB _ à _**] à



Étape 2 Dans l'espace de travail, fais un clic droit sur le bloc [**définir LED RVB _ à _**] et clique sur "Dupliquer". Recommence jusqu'à obtenir 4 blocs [**définir LED RVB _ à _**].



Étape 3 Imbrique-les dans le bloc [**au démarrage**]. Change les numéros de LEDs des trois derniers blocs de 0 en 1, 2 et 3 respectivement.



*Il y a 4 LEDs RVB sur le RGB Bit et chacune d'elle est identifiée par un numéro (de 0 à 3).
Utilise-les pour programmer chaque LED individuellement.*

Étape 4 Flashe le code sur EDU:BIT.



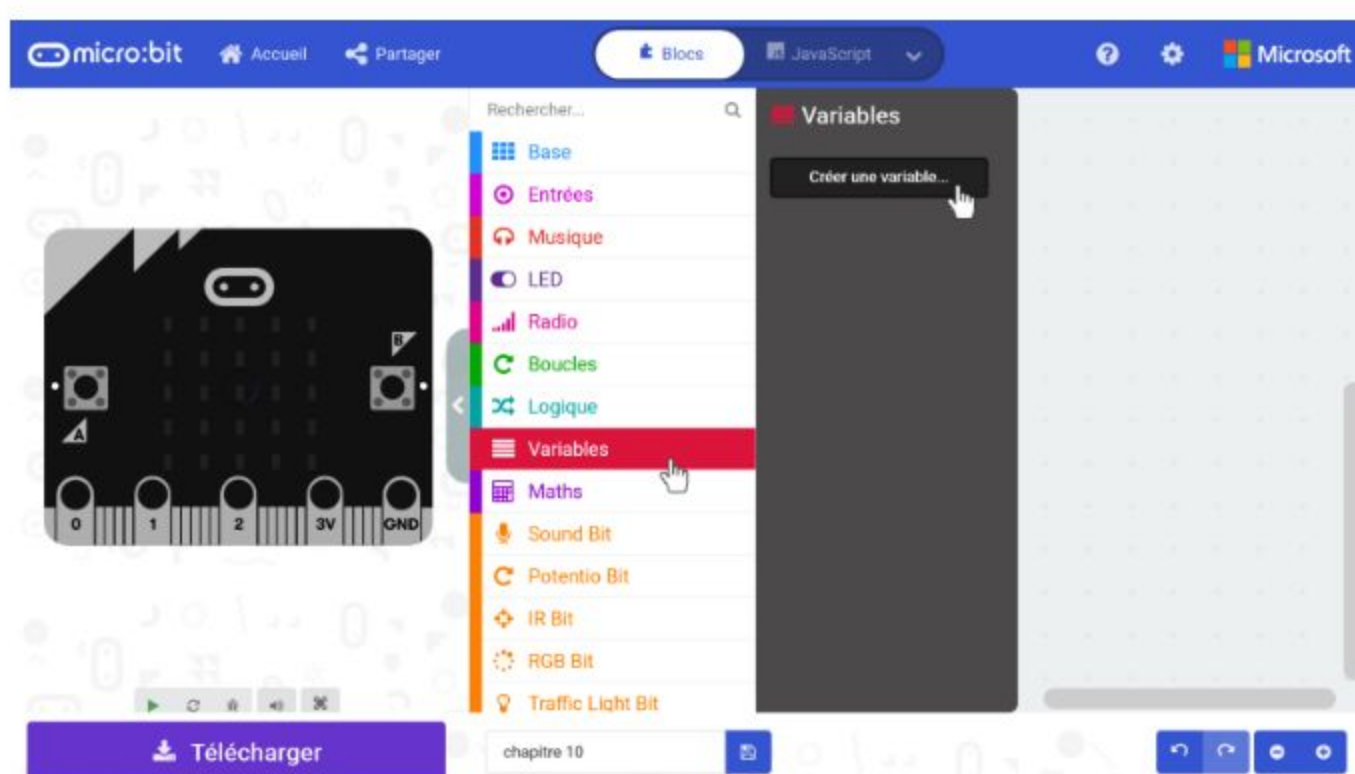
Lorsque tu allumes la carte EDU:BIT, les quatre LEDs du RGB Bit s'allument selon des couleurs définies.

Tu peux facilement changer la couleur d'une LED en cliquant sur la couleur du bloc, puis en choisissant une couleur dans la palette de couleurs qui s'affiche.

Essaie !

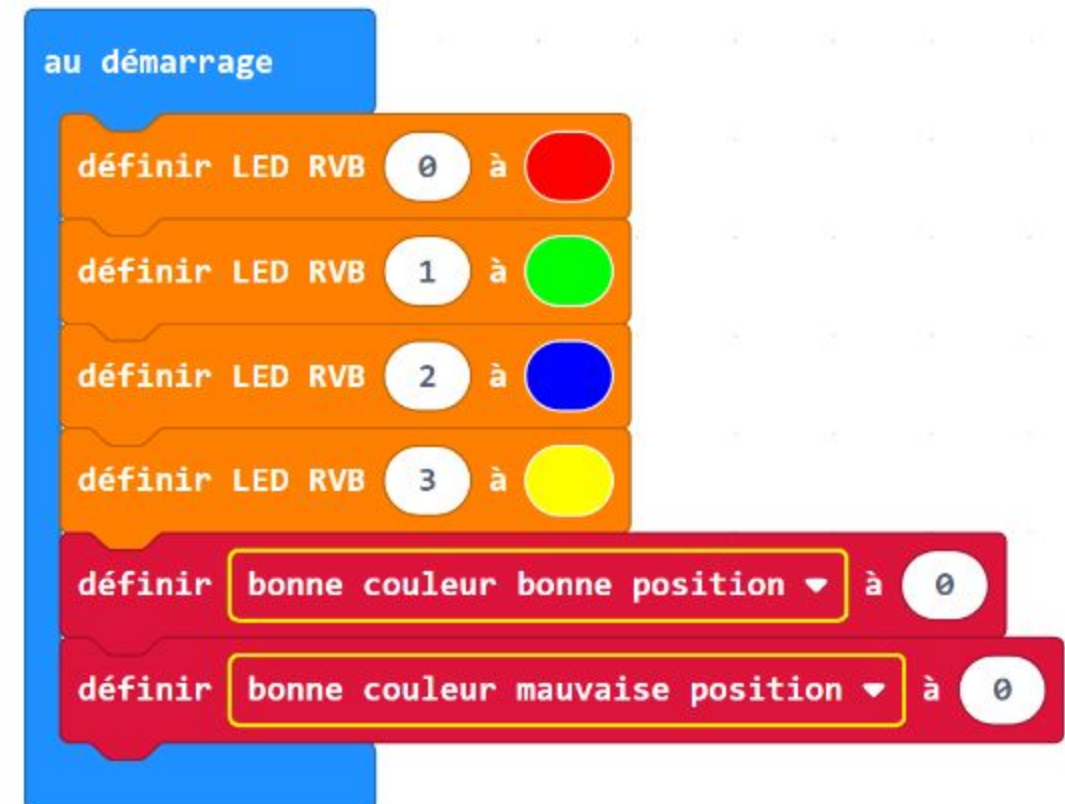
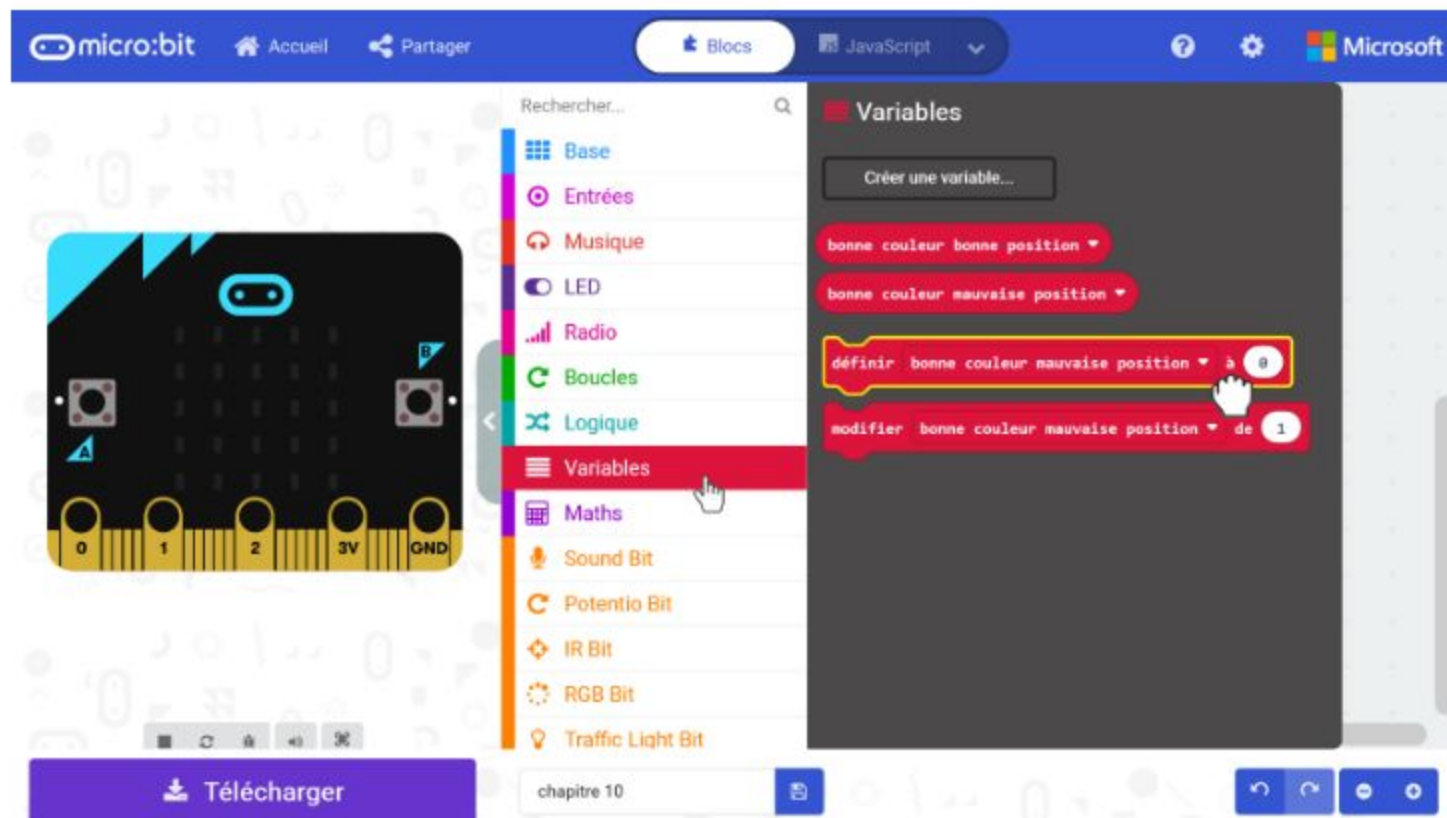


Étape 5 Ajoute deux variables et nomme-les "bonne couleur bonne position" et "bonne couleur mauvaise position".

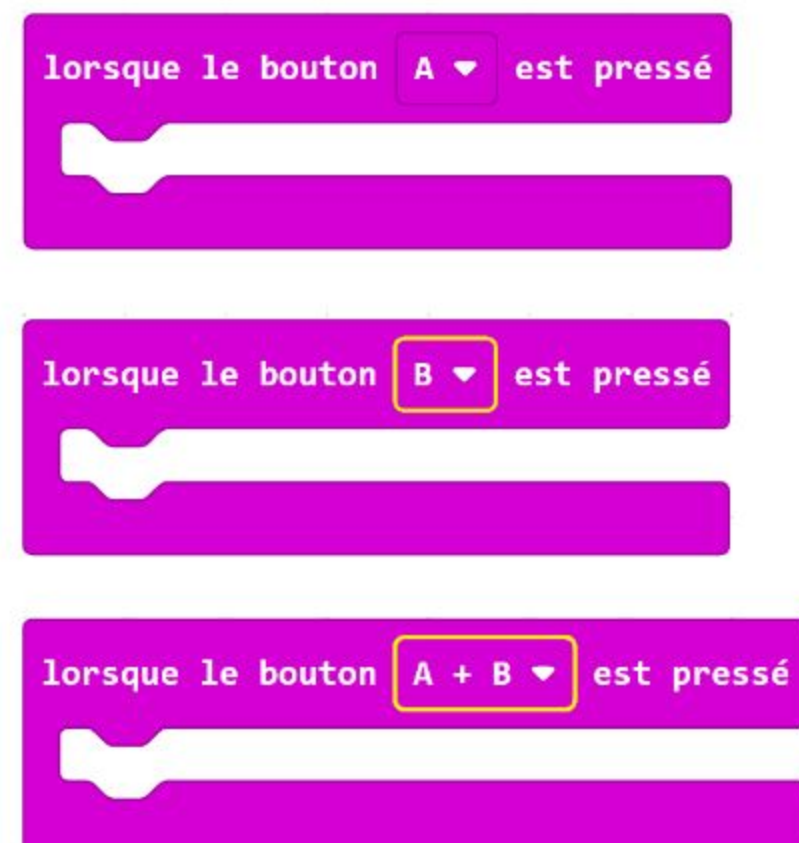
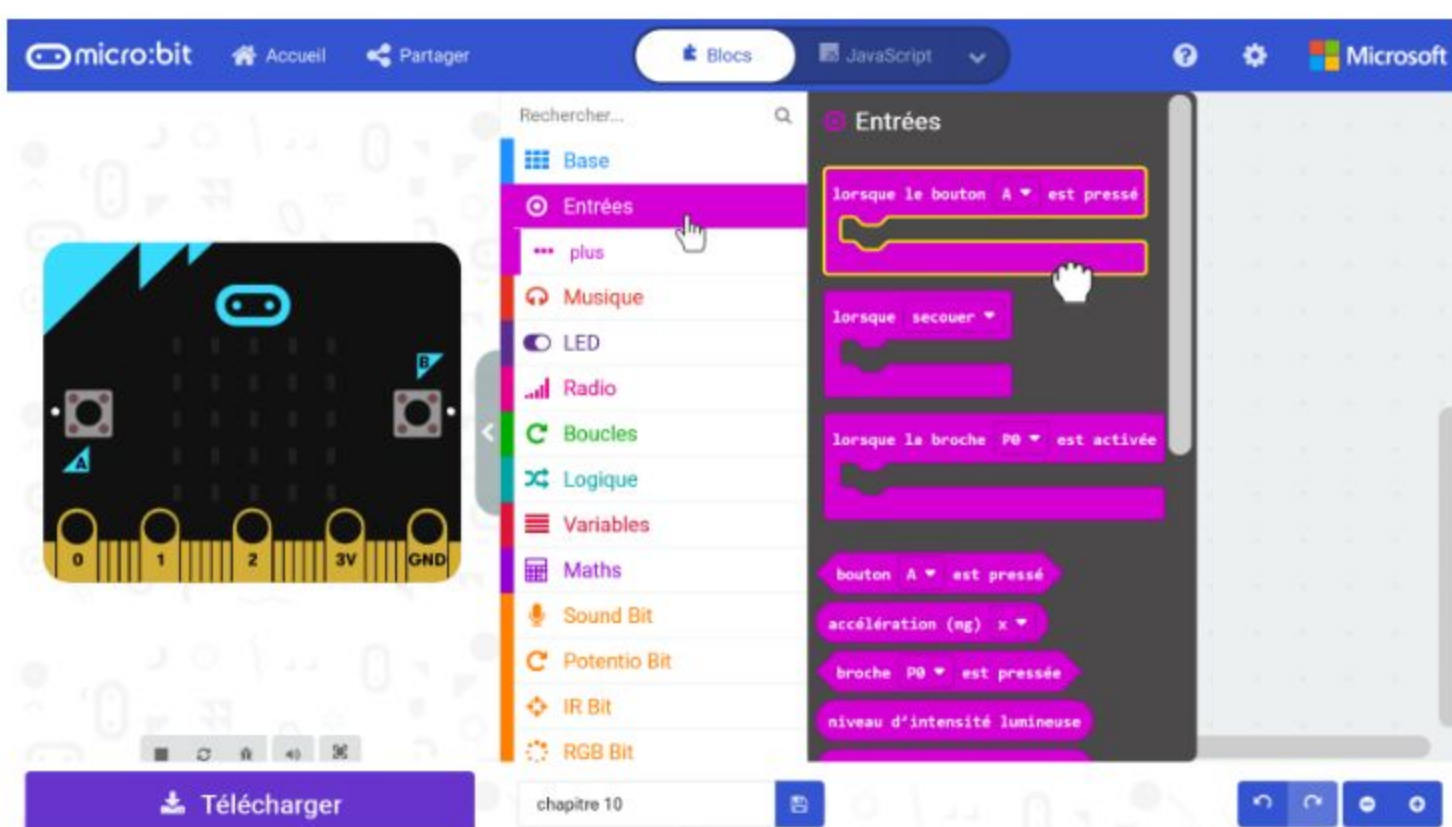




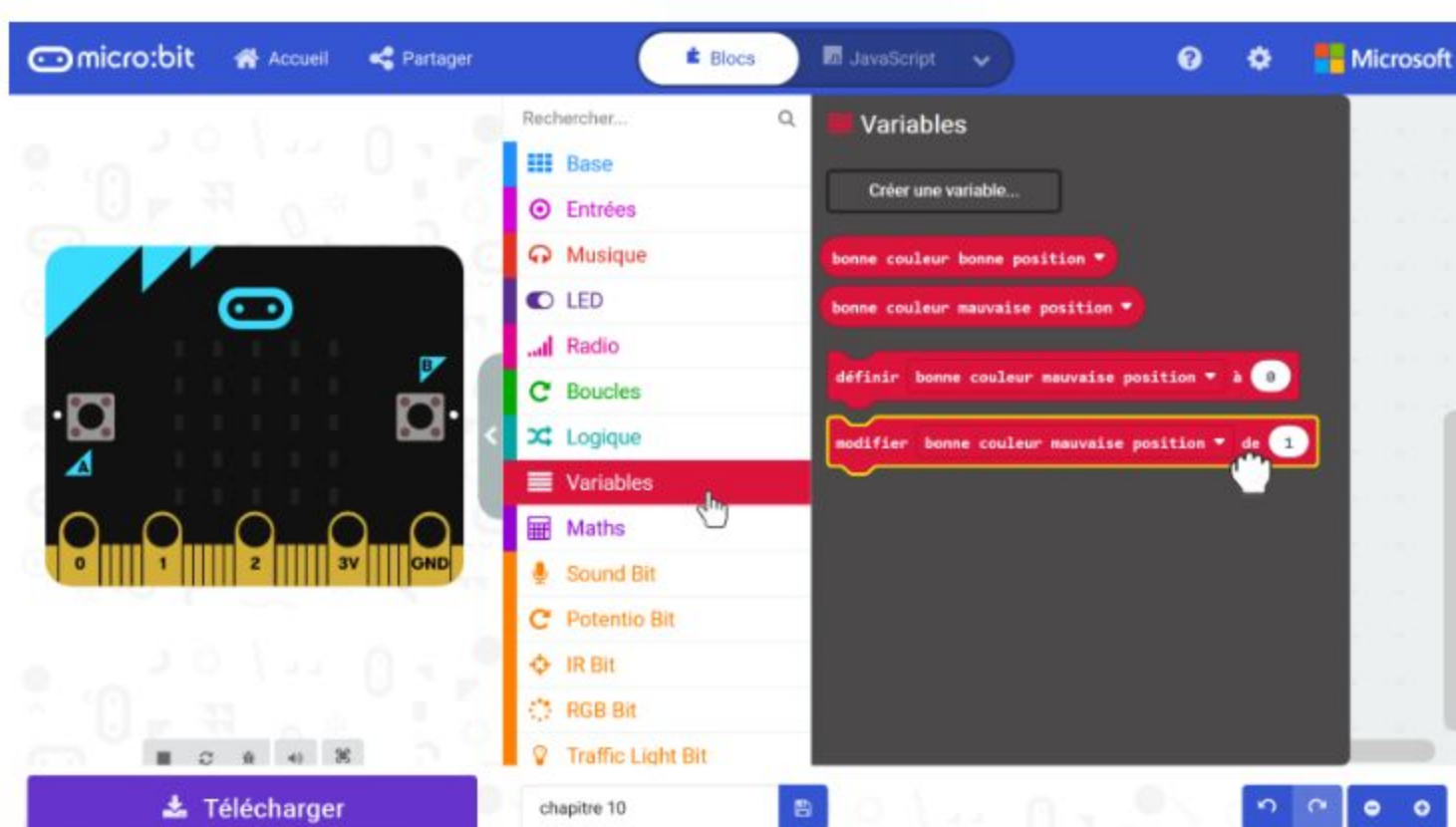
Étape 6 Clique sur la catégorie [**Variables**] et sélectionne le bloc [**définir _ à _**]. Duplique-le et imbrique-les dans le bloc [**au démarrage**]. Sélectionne la variable “bonne couleur bonne position” pour l’un d’eux et “bonne couleur mauvaise position” pour l’autre.



Étape 7 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**]. Duplique-le deux fois, et change les boutons en “B” et “A+B”.

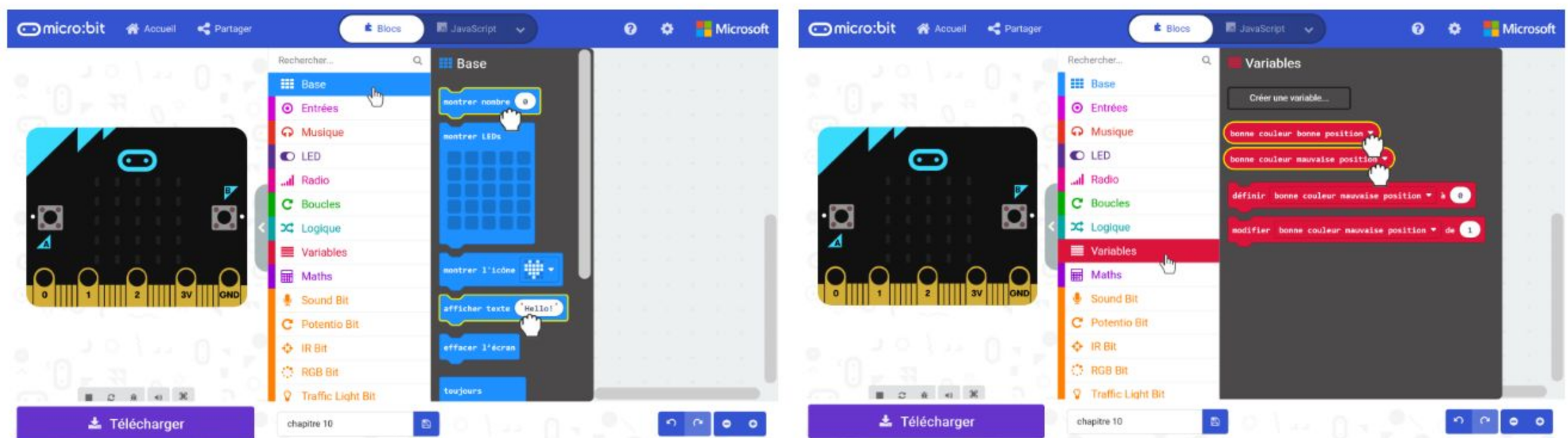


Étape 8 Clique sur la catégorie [**Variables**] et sélectionne le bloc [**modifier _ de _**]. Duplique-le et insère-les dans les blocs [**lorsque le bouton A est pressé**] et [**lorsque le bouton B est pressé**]. Sélectionne les variables “bonne couleur bonne position” pour le premier et “bonne couleur mauvaise position” pour l’autre.




CHAPITRE 10 : Mastermind

Étape 9 Ajoute les blocs [montrer nombre] et [afficher texte] de la catégorie [Base], ainsi que les blocs [bonne couleur bonne position] et [bonne couleur mauvaise position] de la catégorie [Variables]. Insère-les dans les blocs [montrer nombre].



Étape 10 Dans les blocs [afficher texte], change "Hello!" en "A =" et "B =" respectivement. Voici le code complet.



```
au démarrage
  définir LED RVB 0 à [rouge]
  définir LED RVB 1 à [vert]
  définir LED RVB 2 à [bleu]
  définir LED RVB 3 à [jaune]
  définir bonne couleur bonne position à 0
  définir bonne couleur mauvaise position à 0

lorsque le bouton A est pressé
  modifier bonne couleur bonne position de 1
  montrer nombre bonne couleur bonne position

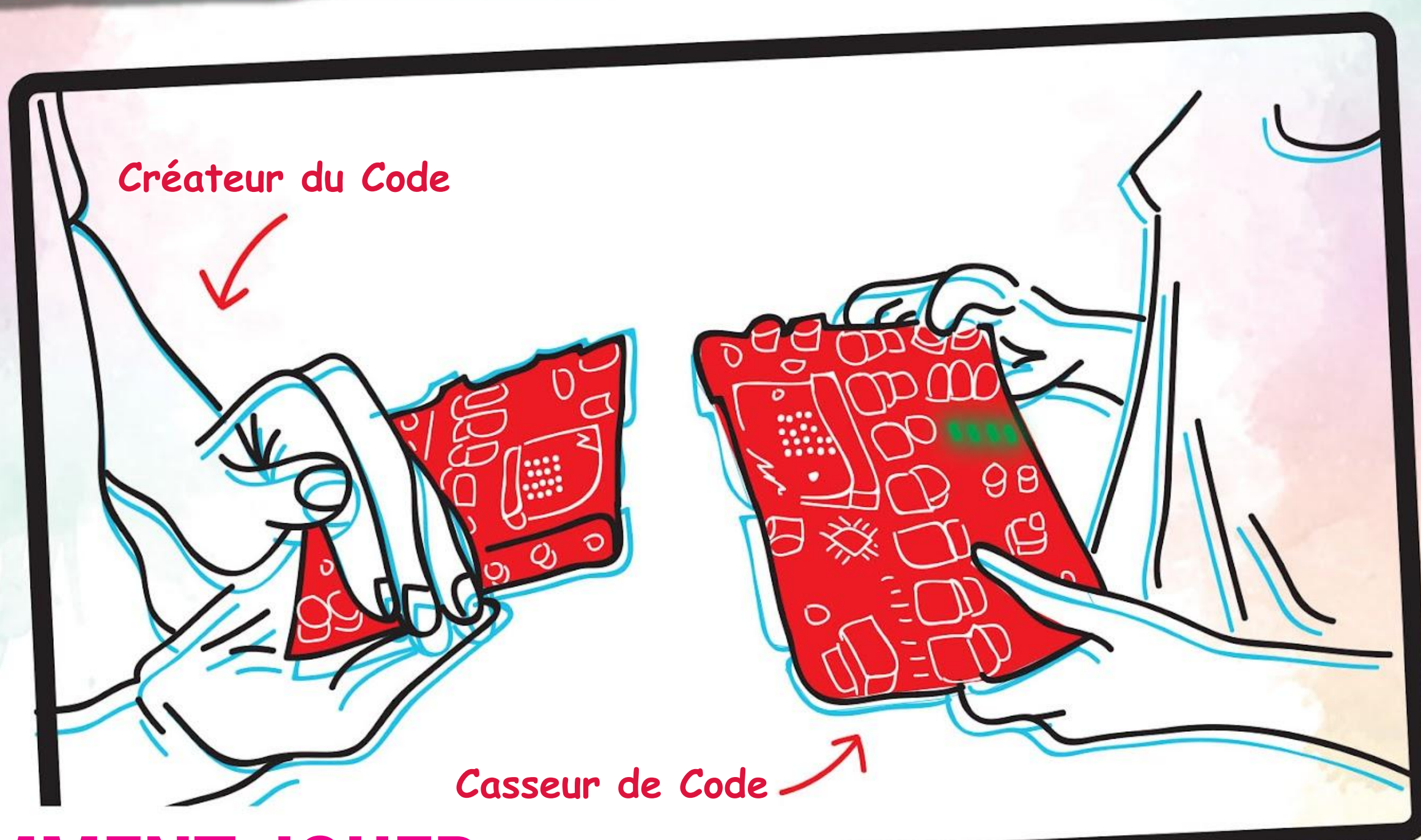
lorsque le bouton B est pressé
  modifier bonne couleur mauvaise position de 1
  montrer nombre bonne couleur mauvaise position

lorsque le bouton A + B est pressé
  afficher texte "A ="
  montrer nombre bonne couleur bonne position
  afficher texte "B ="
  montrer nombre bonne couleur mauvaise position
```

Étape 11 Flashe le code sur EDU:BIT. Tu es maintenant prêt à jouer à Mastermind.

Jouons

Mastermind, Peux-Tu Casser le Code ?



COMMENT JOUER :

- Le joueur 1, créateur du code, définit le code secret en allumant les 4 LEDs aléatoirement. Limiter le choix des couleurs à rouge et vert pour commencer. Penser à cacher la carte EDU:BIT afin que l'autre joueur ne puisse voir la combinaison de couleurs à trouver.
- Le joueur 2, casseur de code, essaie de deviner le code. Le joueur 2 allume les LEDs RVB sur son EDU:BIT, puis montre sa proposition au créateur du code.
- Le créateur du code vérifie et appuie sur le bouton jaune (bouton A) et / ou sur le bouton bleu (bouton B) sur la carte EDU:BIT du casseur de code pour indiquer combien de LED's allumées sont de la "bonne couleur bonne position" et combien sont de la "bonne couleur mauvaise position".
- Le casseur de code appuie ensuite sur les boutons jaune et bleu simultanément pour connaître l'évaluation de sa proposition.
- Répéter les étapes 2 à 4 jusqu'à ce que la bonne combinaison de couleurs soit trouvée (10 essais maximum).
- Échanger les rôles pour jouer une nouvelle manche.

Comment Gagner :

Tu gagnes la partie si tu réussis à déchiffrer (ou casser le code, ie : deviner la bonne combinaison de couleurs). En cas d'échec, la victoire revient au créateur du code.



EXPLORE D'AUTRES BLOCS :

#1 Tu peux programmer les LEDs du RGB Bit pour qu'elles affichent les couleurs de l'arc-en-ciel en remplaçant le bloc [définir LEDs RVB _ à _] par le bloc [afficher arc-en-ciel sur LEDs RVB].

```
au démarrage
  afficher arc-en-ciel sur LEDs RVB
```

#2 Tu peux créer un effet "lumière qui se déplace" en plaçant le bloc [rotation LEDs RVB de _] dans le bloc [toujours]. Pense à ajouter un bloc [pause _ (ms)] pour ralentir le programme et ainsi voir l'effet. Voici un exemple de code.

```
au démarrage
  définir LED RVB 0 à [rouge]
  définir LED RVB 1 à [vert]
  définir LED RVB 2 à [bleu]
  définir LED RVB 3 à [jaune]

toujours
  rotation LEDs RVB de 1
  pause (ms) 500
```

#3 Tu peux aussi décaler les lumières sur les LEDs en imbriquant le bloc [translation LEDs RVB de _] dans le bloc [toujours]. Tu auras besoin d'ajouter un bloc [pause _ (ms)] pour ralentir le programme et ainsi voir l'effet. Les lumières se décalent et les LEDs s'éteignent une par une. Voici un exemple de code :

```
au démarrage
  définir LED RVB 0 à [rouge]
  définir LED RVB 1 à [vert]
  définir LED RVB 2 à [bleu]
  définir LED RVB 3 à [jaune]

toujours
  translation LEDs RVB de 1
  pause (ms) 500
```

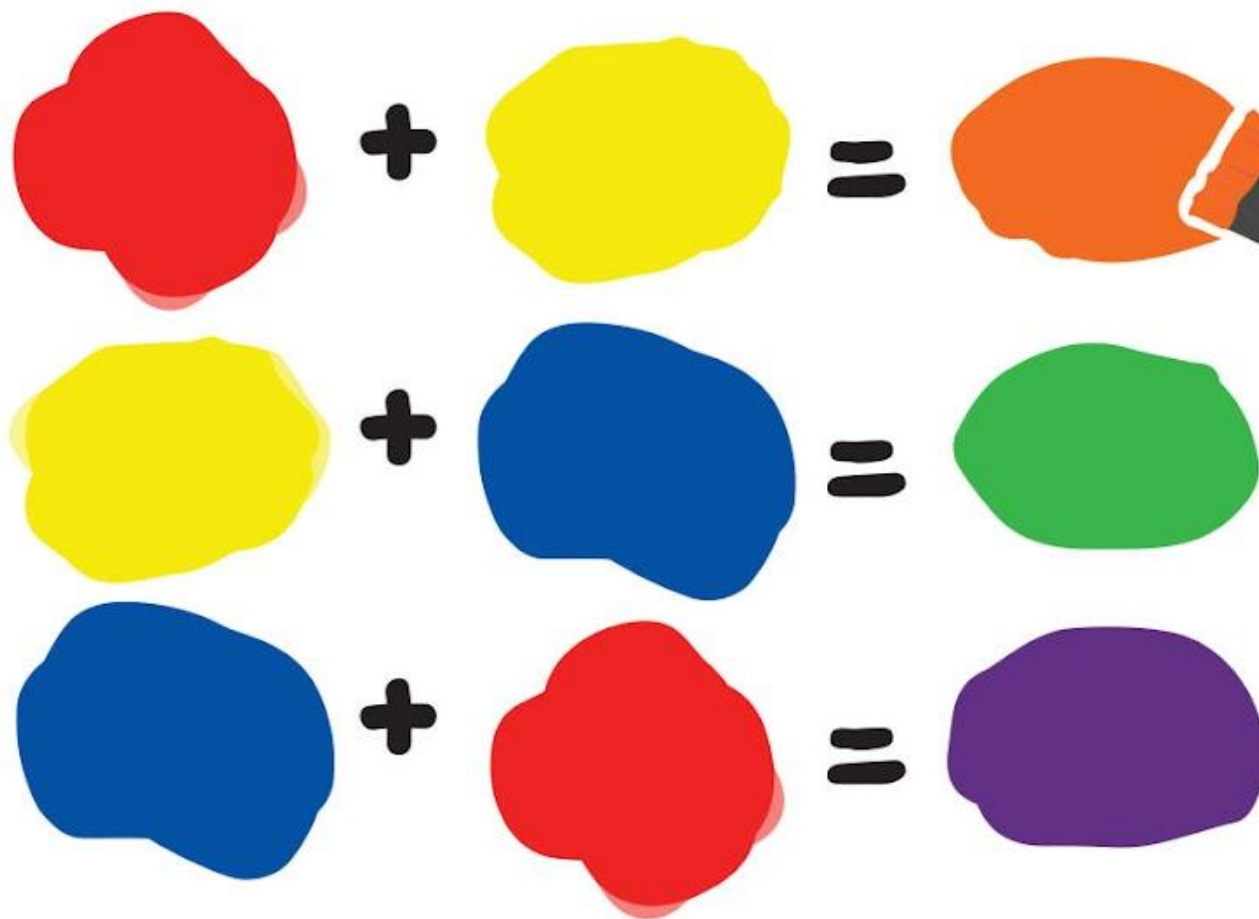


Tu peux changer la direction de l'effet pour #2 et #3 ci-dessus en changeant les valeurs positives en valeurs négatives.

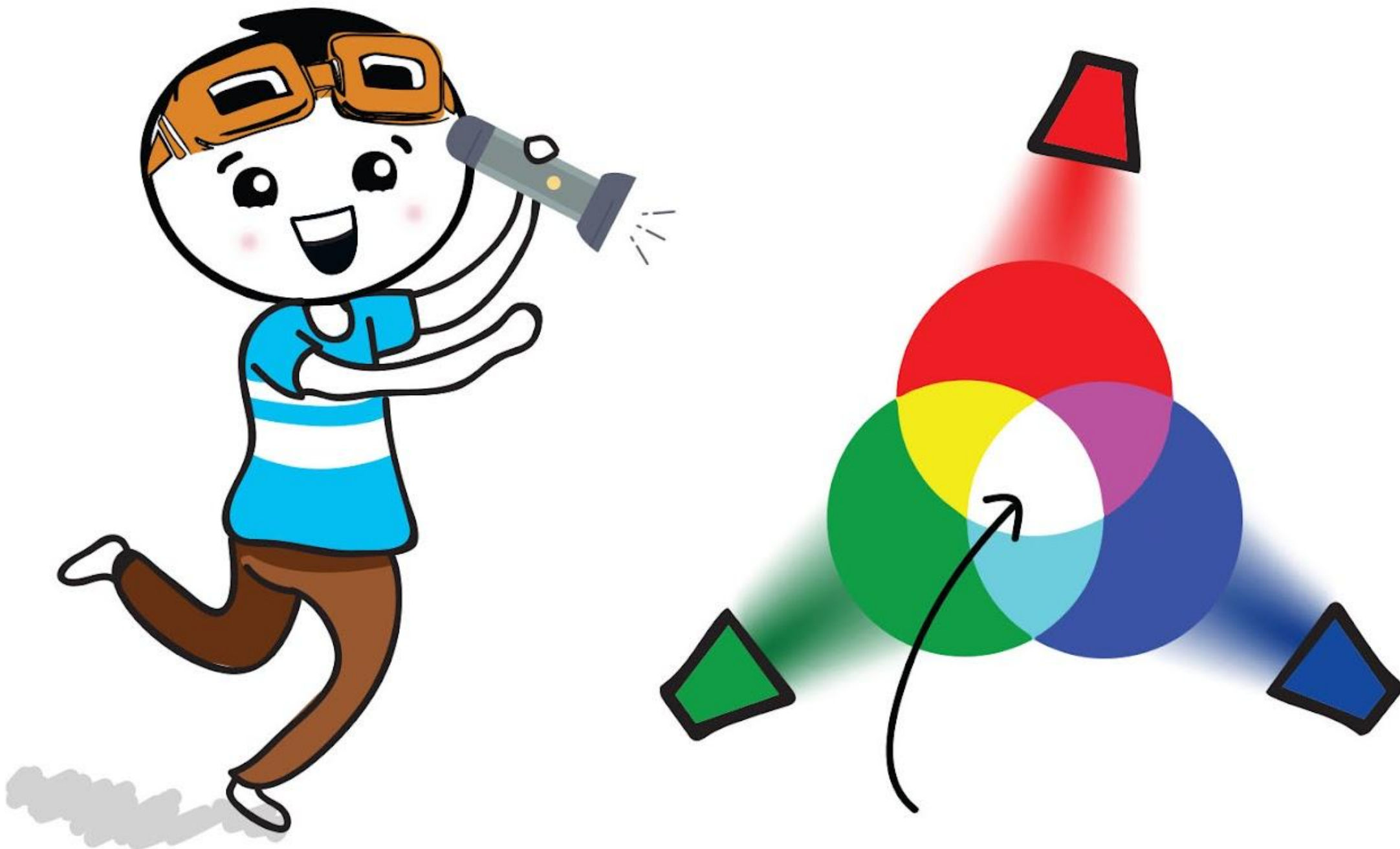
FAIT AMUSANT !



En cours d'arts plastiques, tu as probablement appris que les couleurs primaires étaient Rouge, Jaune et Bleu. Et lorsque tu les mélanges, tu obtiens ces résultats :



Cependant, pour les appareils qui utilisent la lumière pour afficher des couleurs, comme par exemple une télévision ou un écran d'ordinateur, le modèle RVB est utilisé.



Dans ce modèle, les trois lumières primaires sont Rouge (R), Vert (V) et Bleu (B). Les trois ensemble donnent une lumière blanche !

APPLICATION DÉFI

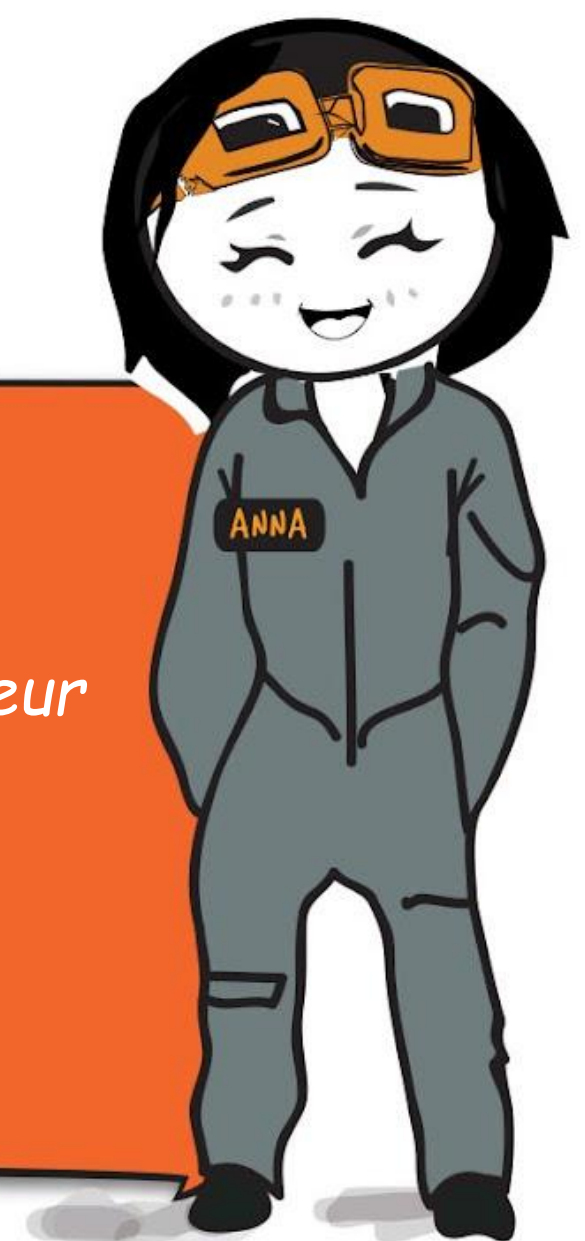
Programme EDU:BIT pour qu'il serve d'outil ludique pour exercer sa mémoire.

Comment ça marche ?

- Pour commencer, incliner EDU:BIT vers la gauche pour qu'elle affiche aléatoirement une combinaison de couleurs sur les LEDs du RGB Bit pendant quelques secondes.
- Tu dois observer et mémoriser le motif (ie : la combinaison de couleurs), puis le redonner oralement une fois les LEDs éteintes.
- Pour vérifier la réponse, appuyer sur le bouton bleu (bouton B) afin d'allumer les LEDs RVB, et donc réafficher le motif.
- Si la réponse est bonne, appuyer sur le bouton jaune (bouton A) pour incrémenter et afficher le score. Sinon, la partie est finie.
- Il est possible de régler la difficulté du jeu en tournant le bouton du Potentio Bit pour augmenter / diminuer la durée d'affichage du motif à mémoriser.
- Le joueur dont le score est le plus élevé gagne.

Voici quelques conseils pour toi :

*#1: Tu as besoin de créer deux variables : score et motif.
#2: Tu devras prédéfinir la combinaison de couleurs (couleur pour chaque LED RVB) pour chaque motif. Utilise plus de couleurs ou prédéfinis plus de motifs pour augmenter la difficulté. Et inversement, tu peux limiter le nombre de couleurs / de motifs pour les joueurs les plus jeunes.*



CHAPITRE BONUS

Jacques a dit, avec LEDs

Communication Radio

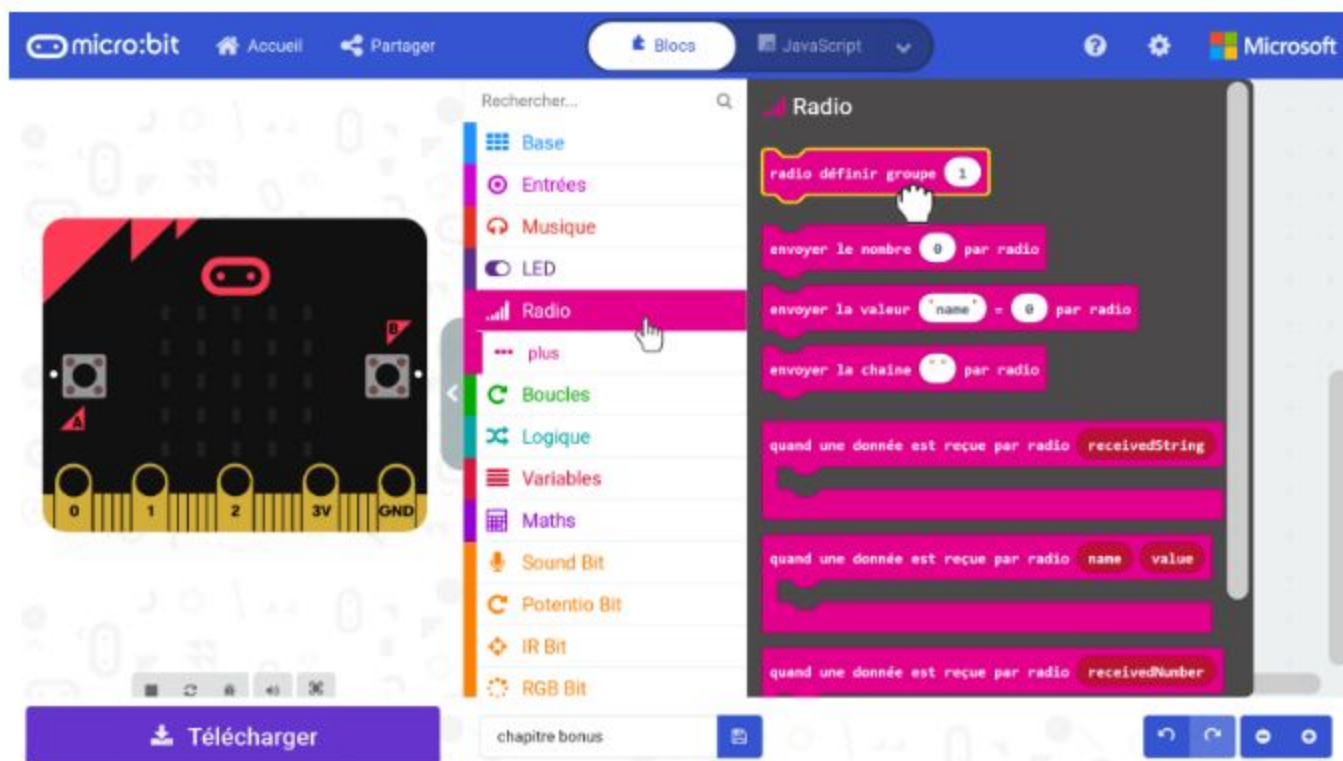




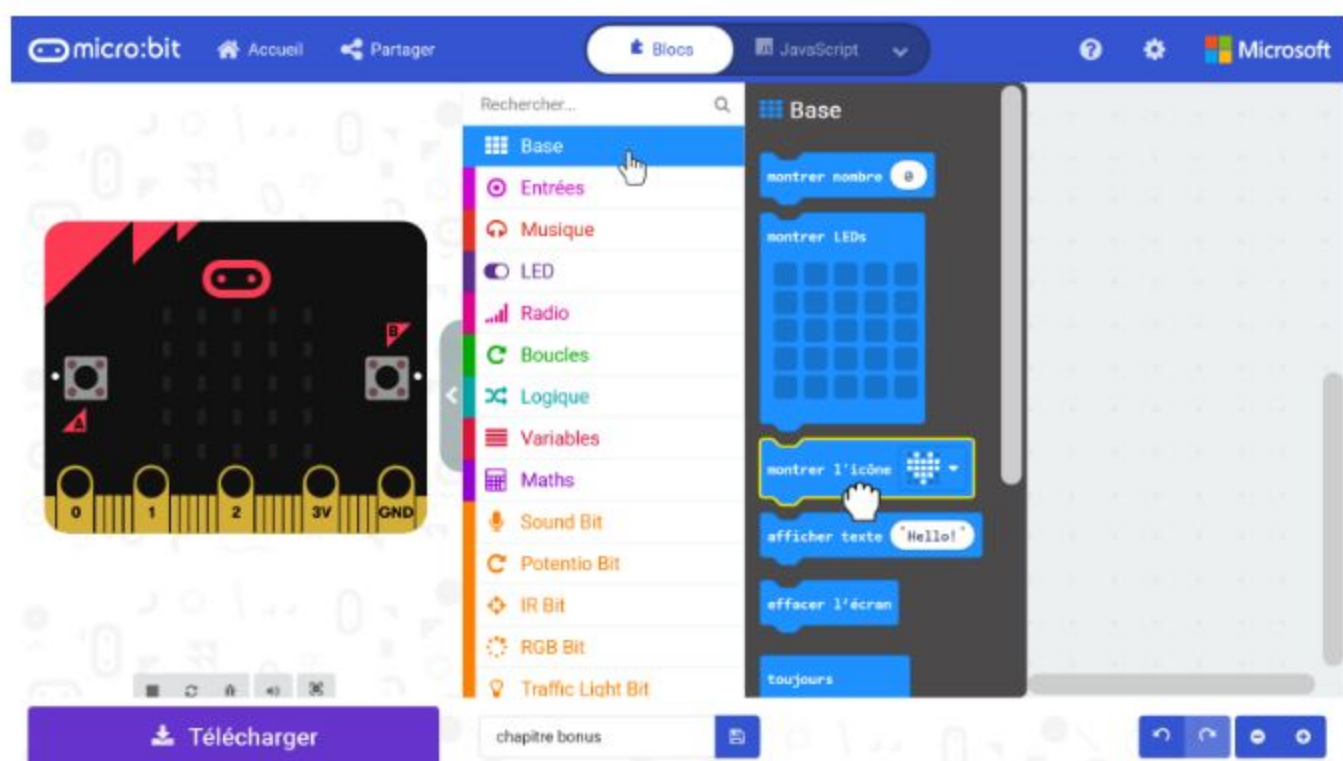
Pour qu'une forme de communication puisse avoir lieu, il faut au moins deux parties : un émetteur et un récepteur. Dans ce jeu, nous avons besoin de deux EDU:BITs qui communiqueront l'une avec l'autre en émettant et recevant des ondes radio.

CODONS !

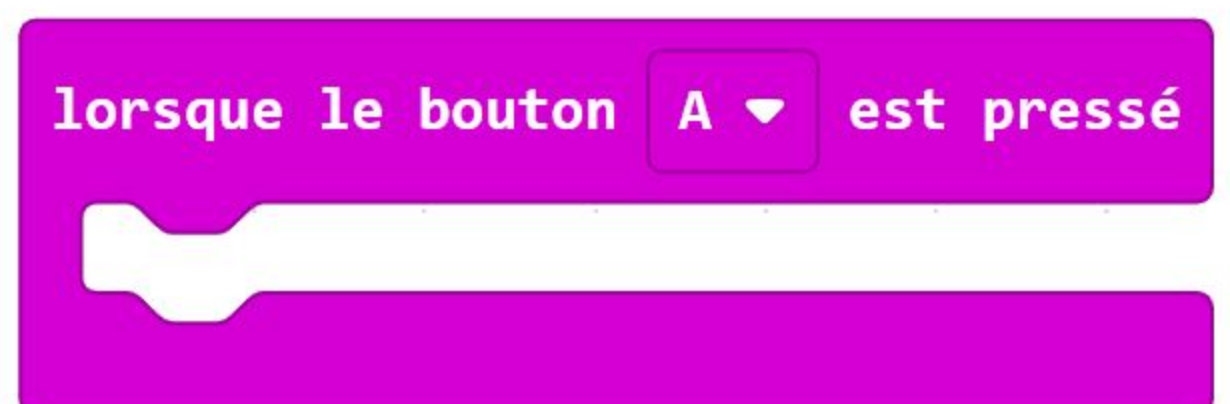
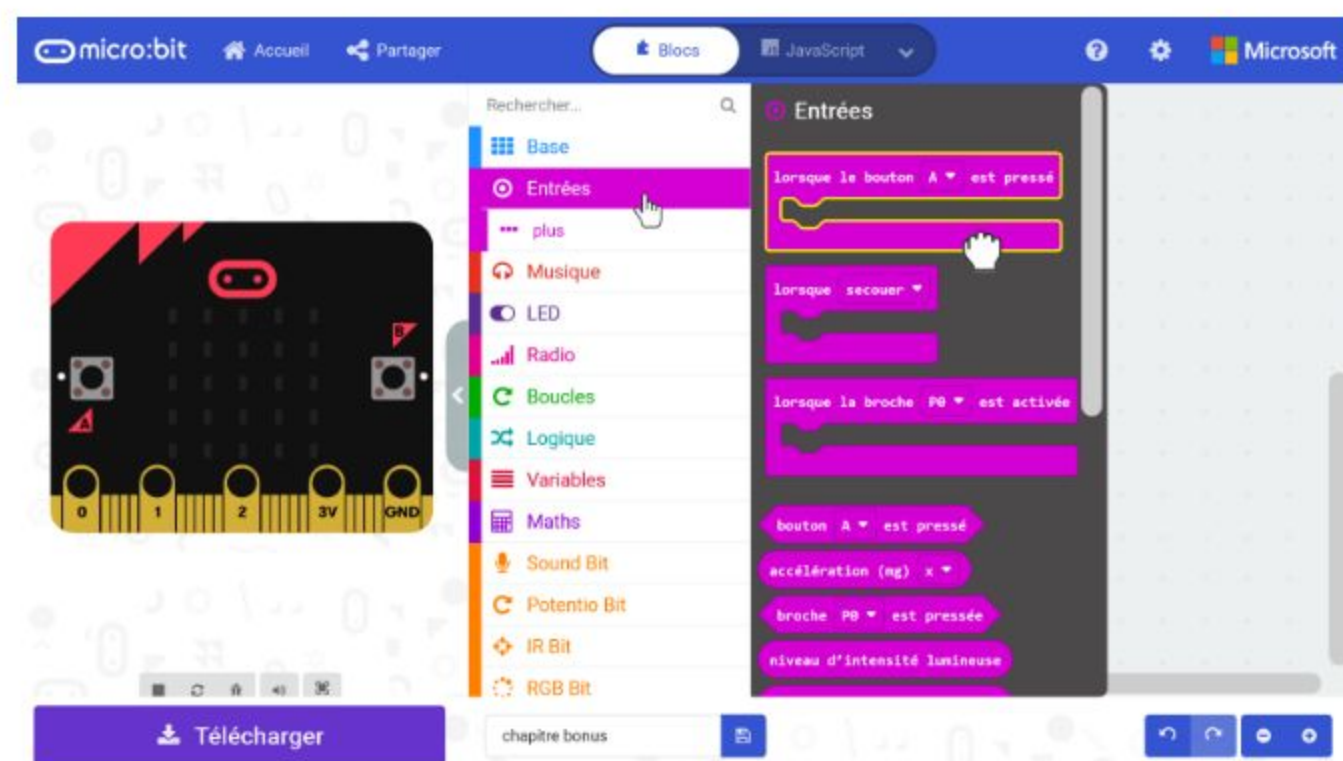
Étape 1 Crée un nouveau projet dans l'éditeur MakeCode et ajoute l'extension EDU:BIT. Clique sur la catégorie [**Radio**] et sélectionne le bloc [**radio définir groupe _**]. Imbrique-le dans le bloc [**au démarrage**].



Étape 2 Dans ton éditeur MakeCode, clique sur la catégorie [**Base**] puis ajoute un bloc [**montrer l'icône**] à ton programme.

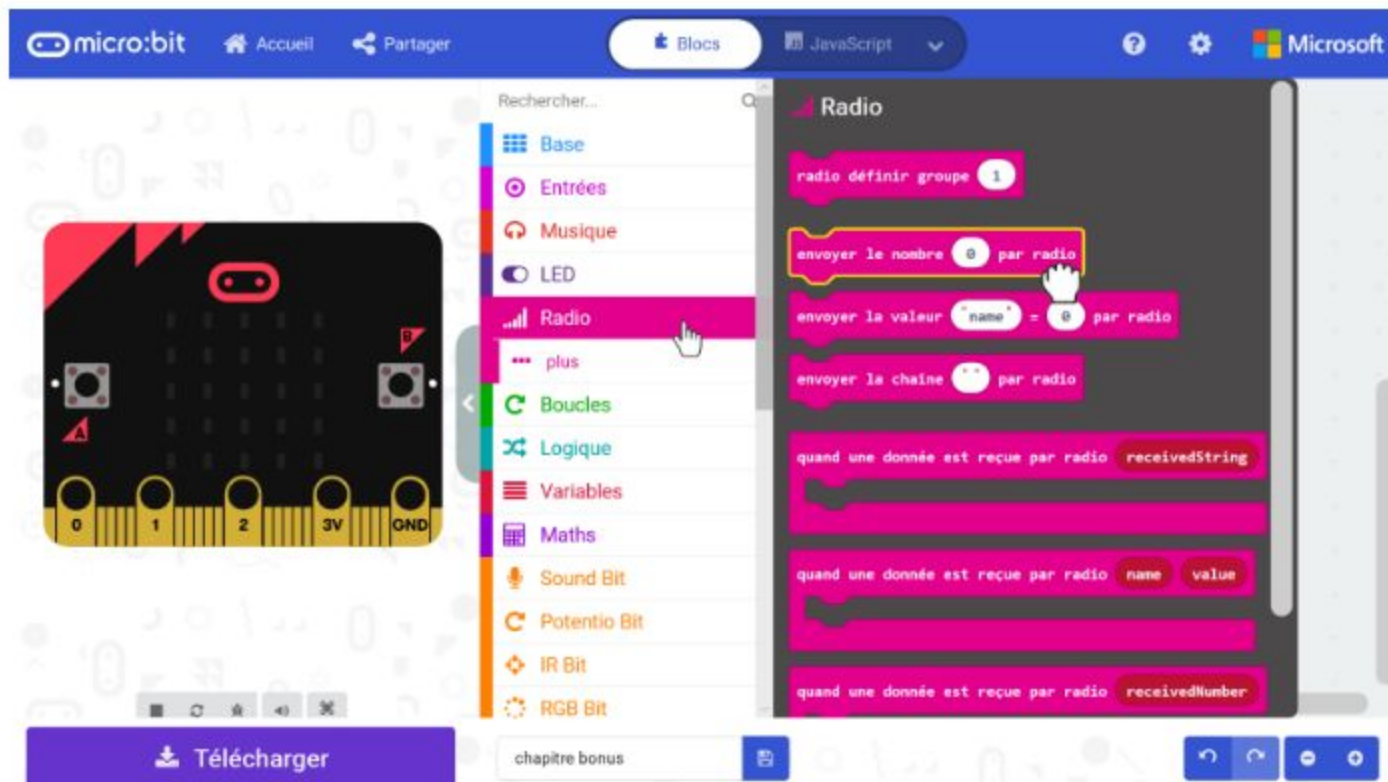


Étape 3 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque le bouton _ est pressé**].

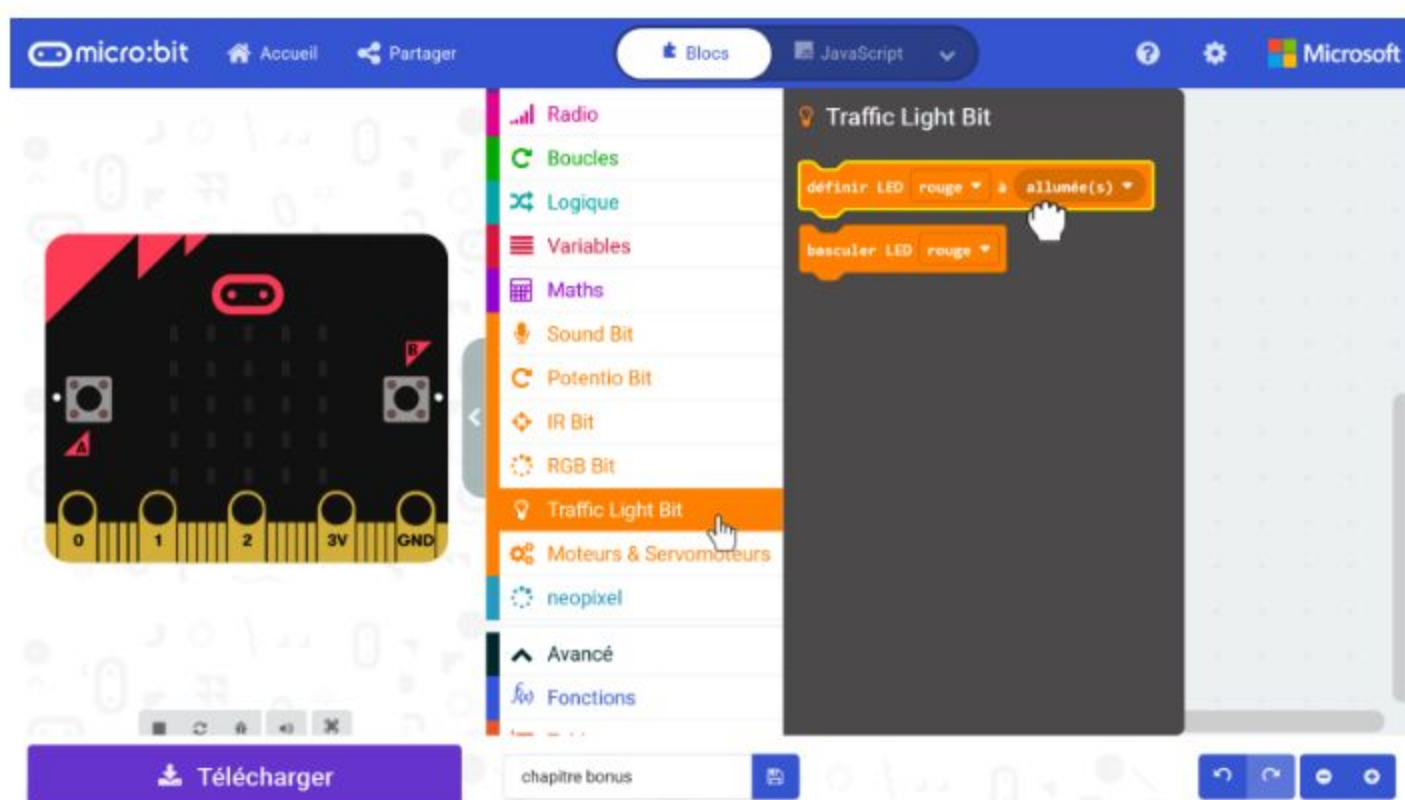




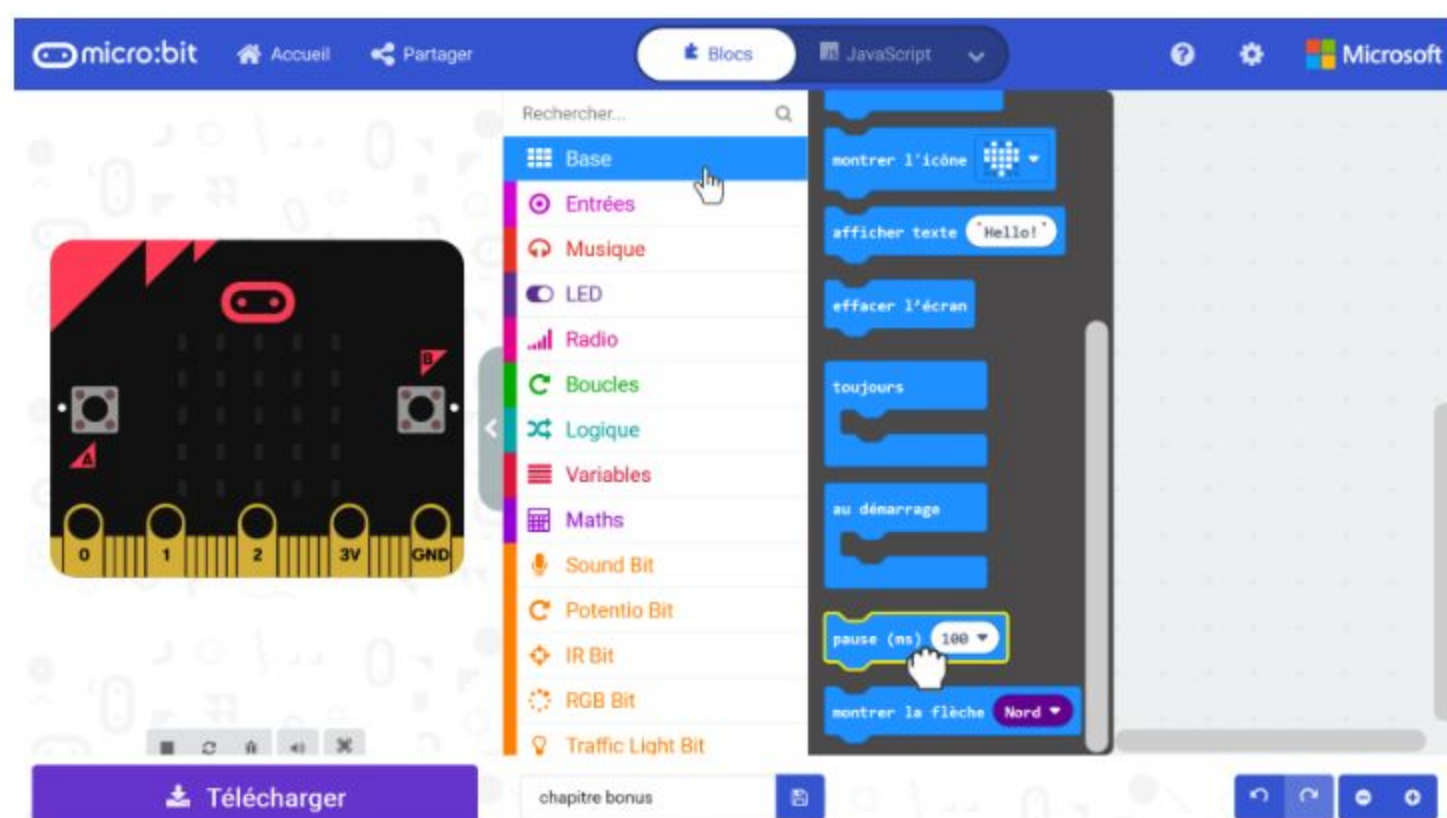
Étape 4 Clique sur la catégorie [**Radio**] et sélectionne le bloc [**envoyer le nombre _ par radio**]. Change la valeur en 1.



Étape 5 Clique sur la catégorie [**Traffic Light Bit**] et sélectionne le bloc [**définir LED _ à _**]. Duplique-le et imbrique-les dans le bloc [**lorsque le bouton A est pressé**]. Change l'état de la LED du deuxième bloc en "éteinte(s)".



Étape 6 Clique sur la catégorie [**Base**] et sélectionne le bloc [**pause _ (ms)**]. Intercalle-le entre les deux blocs [**définir LED _ à _**] et change la valeur en 500.



CHAPITRE BONUS : Jacques a dit, avec LEDs

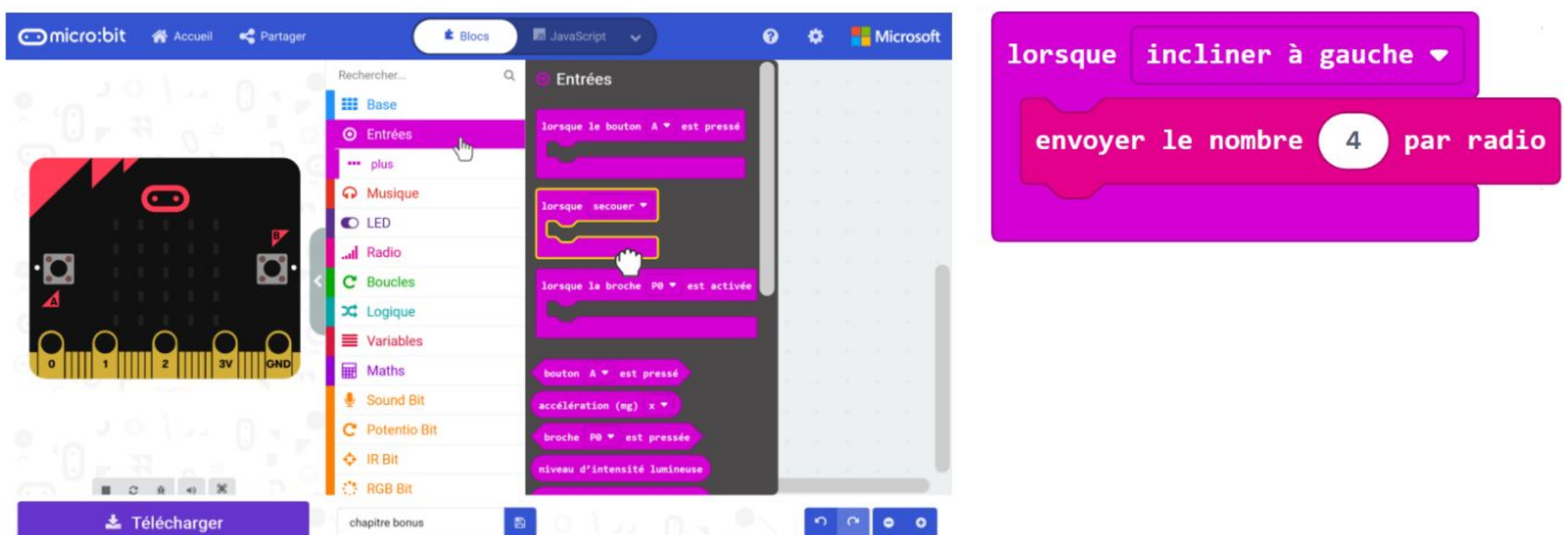
Étape 7 Fais un clic droit sur le bloc [**lorsque le bouton _ est pressé**] et sélectionne "Dupliquer". Répète l'opération pour obtenir trois groupes de blocs identiques.



Étape 8 Change les paramètres suivants : le bouton, le nombre à envoyer et la couleur de la LED des blocs dupliqués comme indiqué ci-dessous :

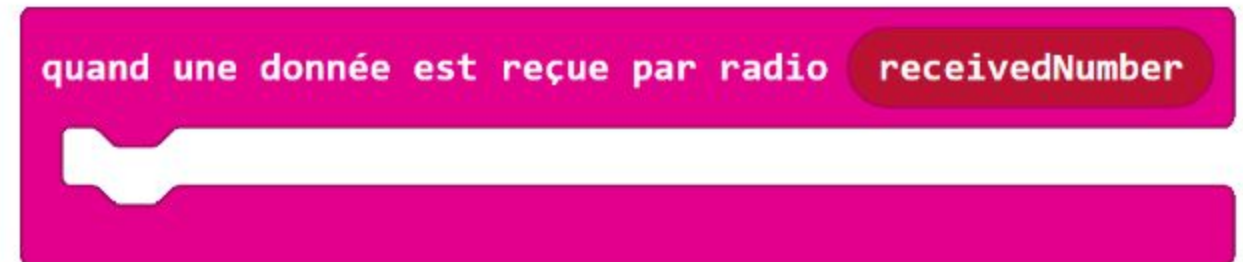
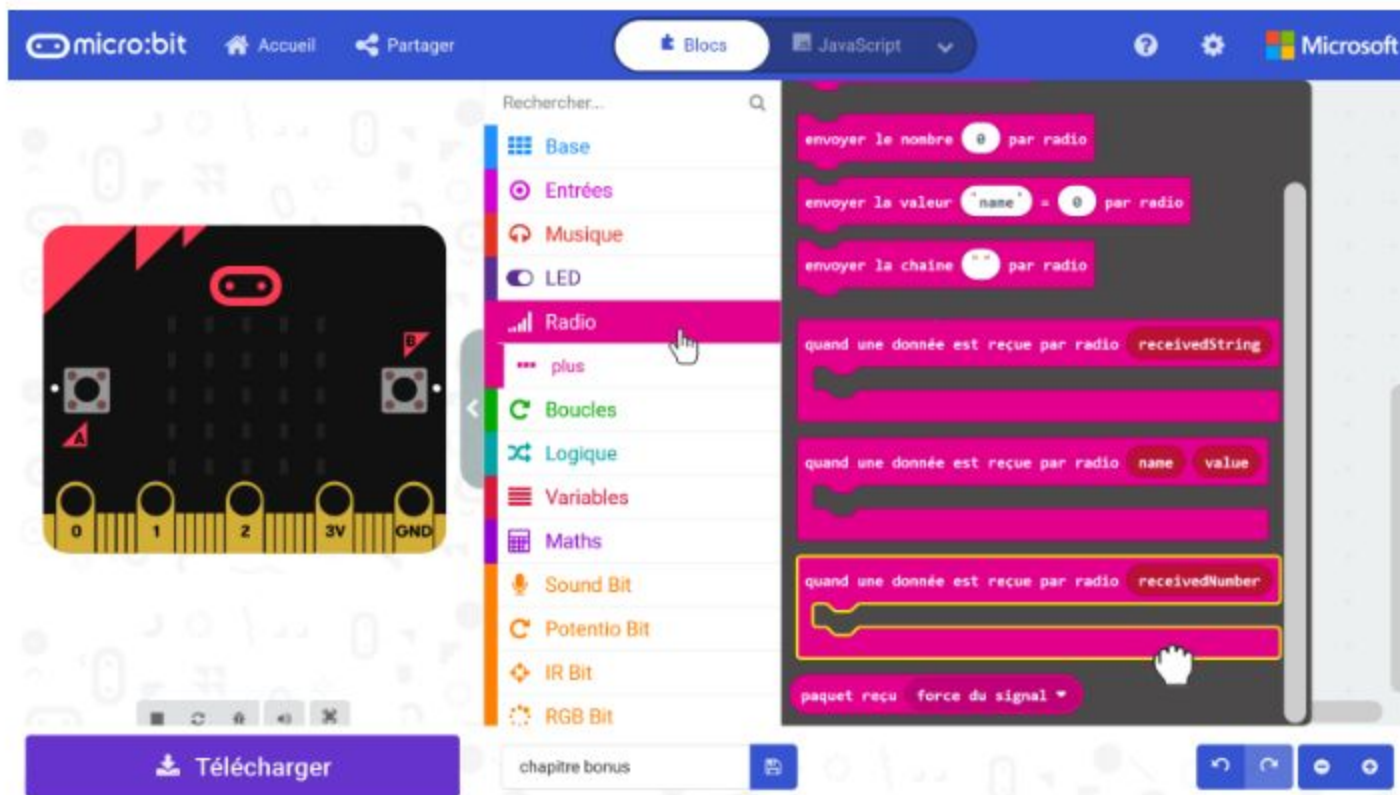


Étape 9 Clique sur la catégorie [**Entrées**] et sélectionne le bloc [**lorsque _**]. Change le paramètre en "incliner à gauche". Clique sur la catégorie [**Radio**] et sélectionne le bloc [**envoyer nombre _ par radio**]. Change la valeur en 4.

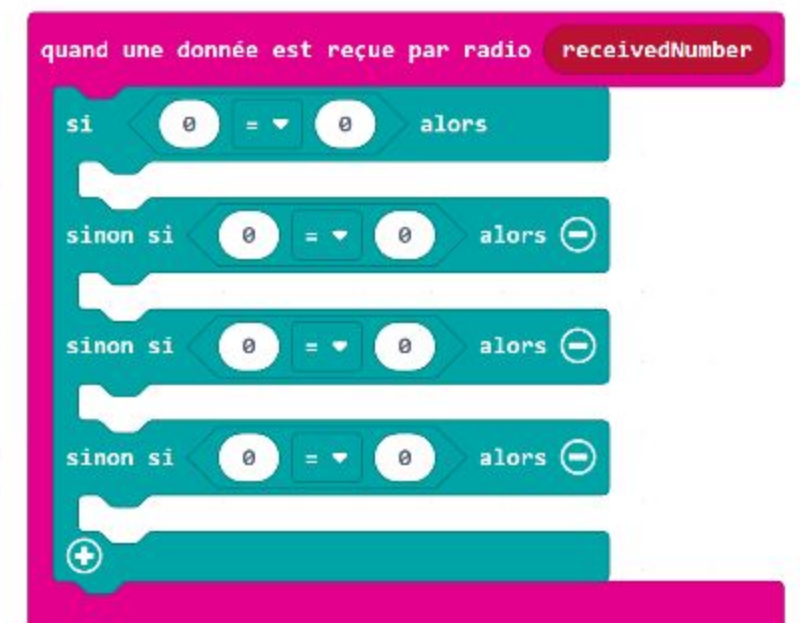




Étape 10 Clique sur la catégorie [**Radio**] et sélectionne le bloc [**quand une donnée est reçue par radio receivedNumber**].



Étape 11 Clique sur la catégorie [**Logique**] et sélectionne le bloc [**si-alors-sinon**]. Clique sur le bouton **+** pour ajouter trois tests [**si**] et clique sur le bouton **-** pour retirer le test [**sinon**]. Insère un bloc de comparaison [**Logique**] : [**_ = _**] dans la condition "si" et chaque condition "sinon si".

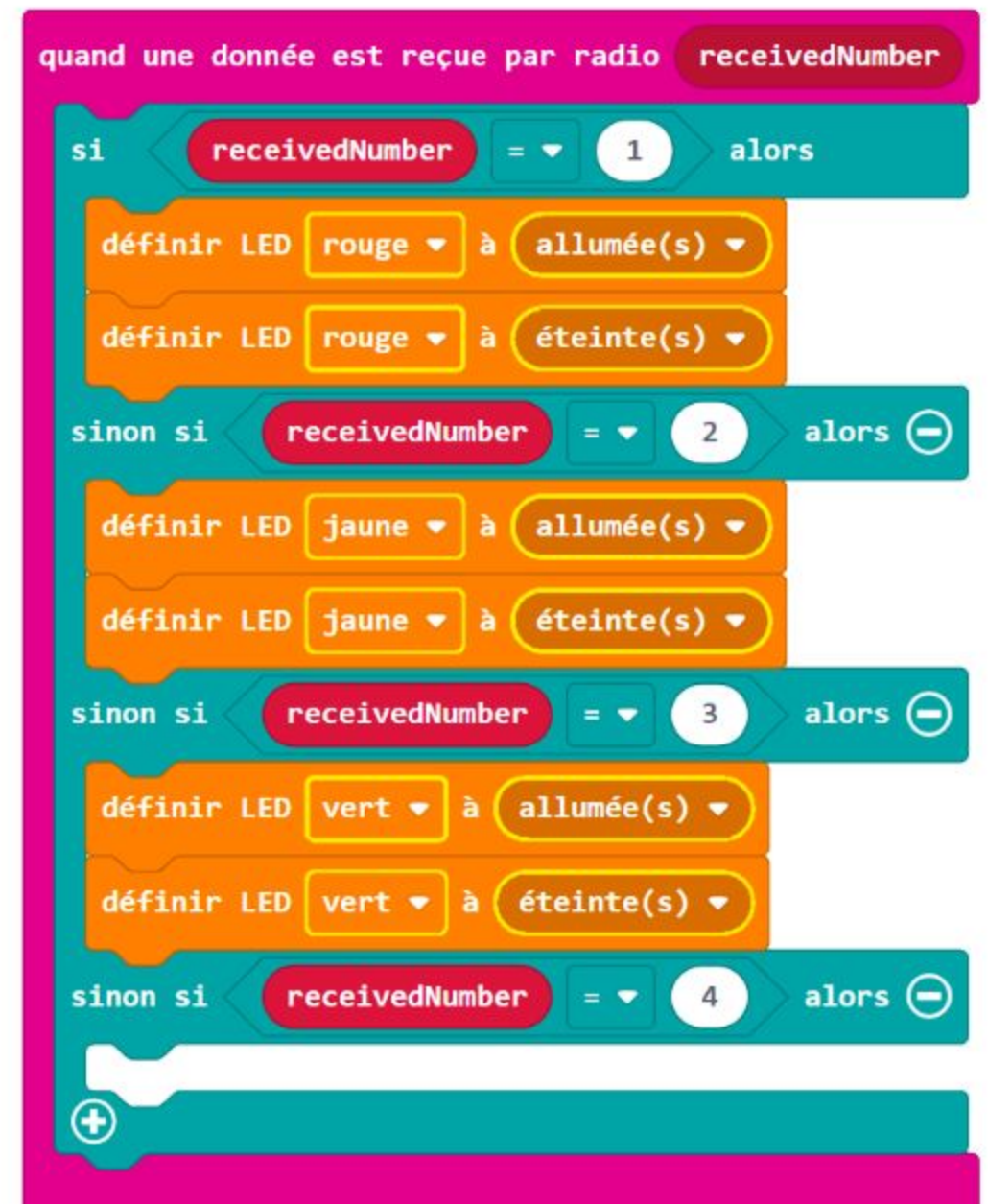
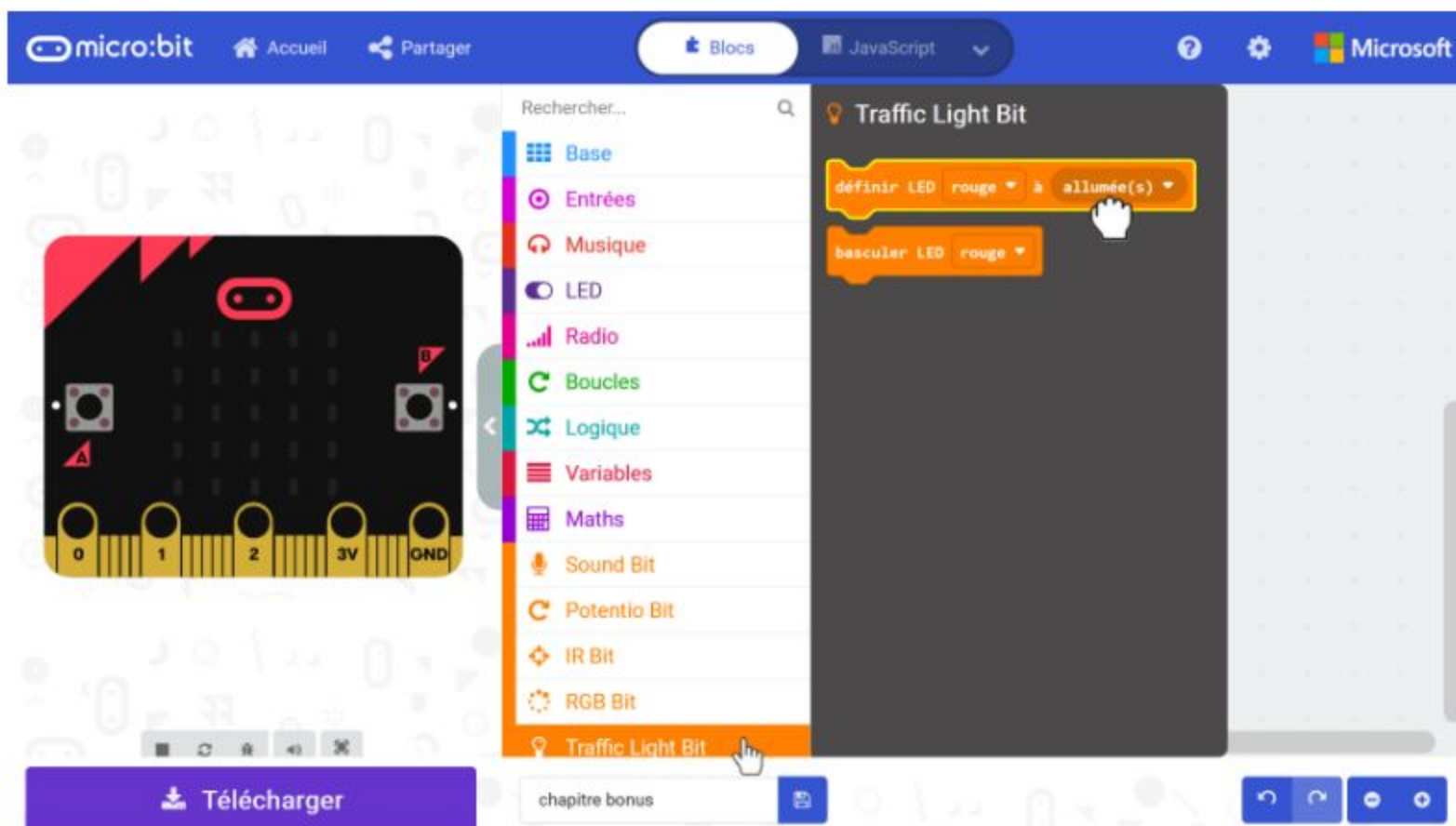


Étape 12 Fais un glisser-déposer de la variable "**receivedNumber**" dans chaque bloc de comparaison (voir ci-dessous) et change les valeurs en **1**, **2**, **3** et **4** respectivement.

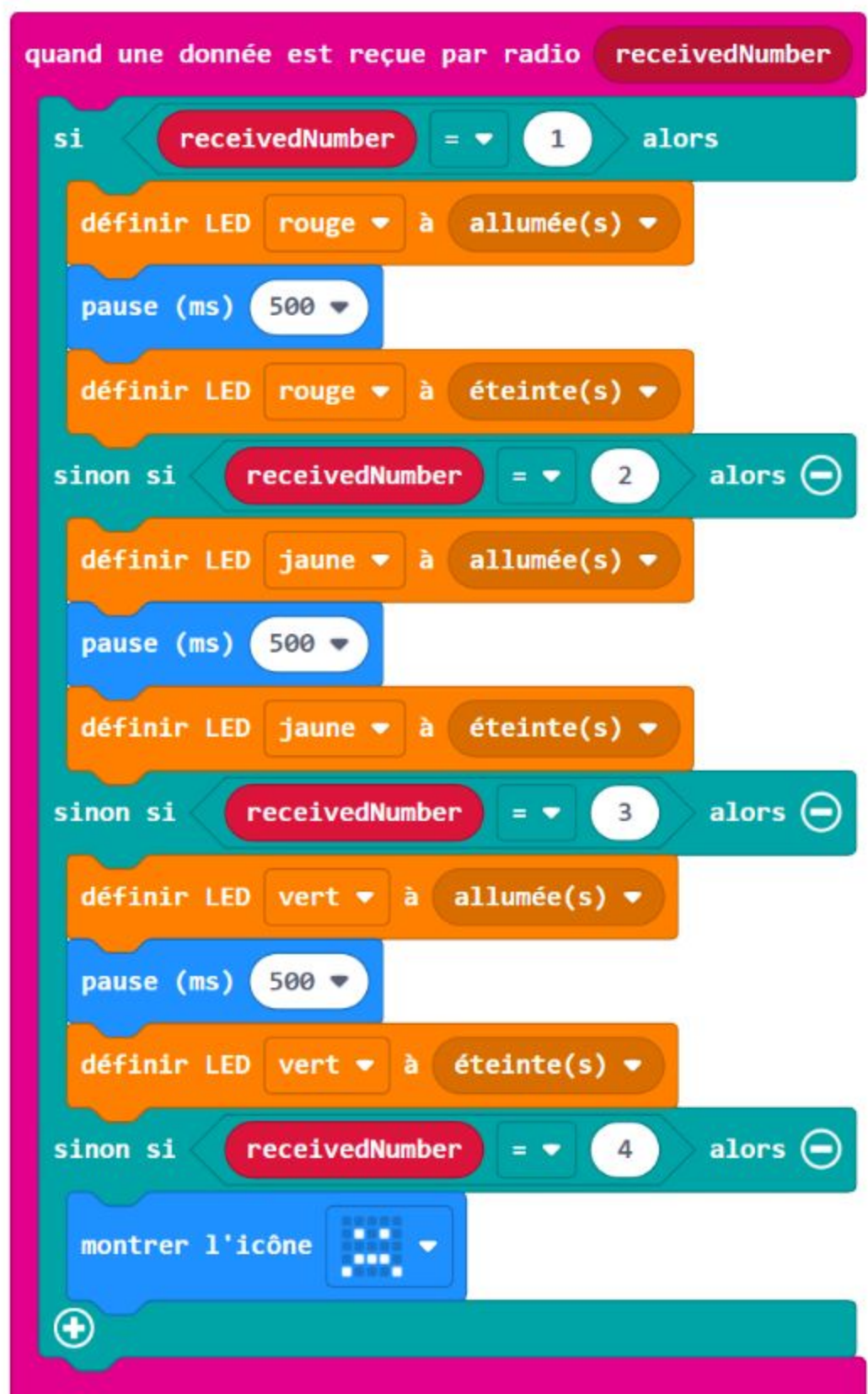
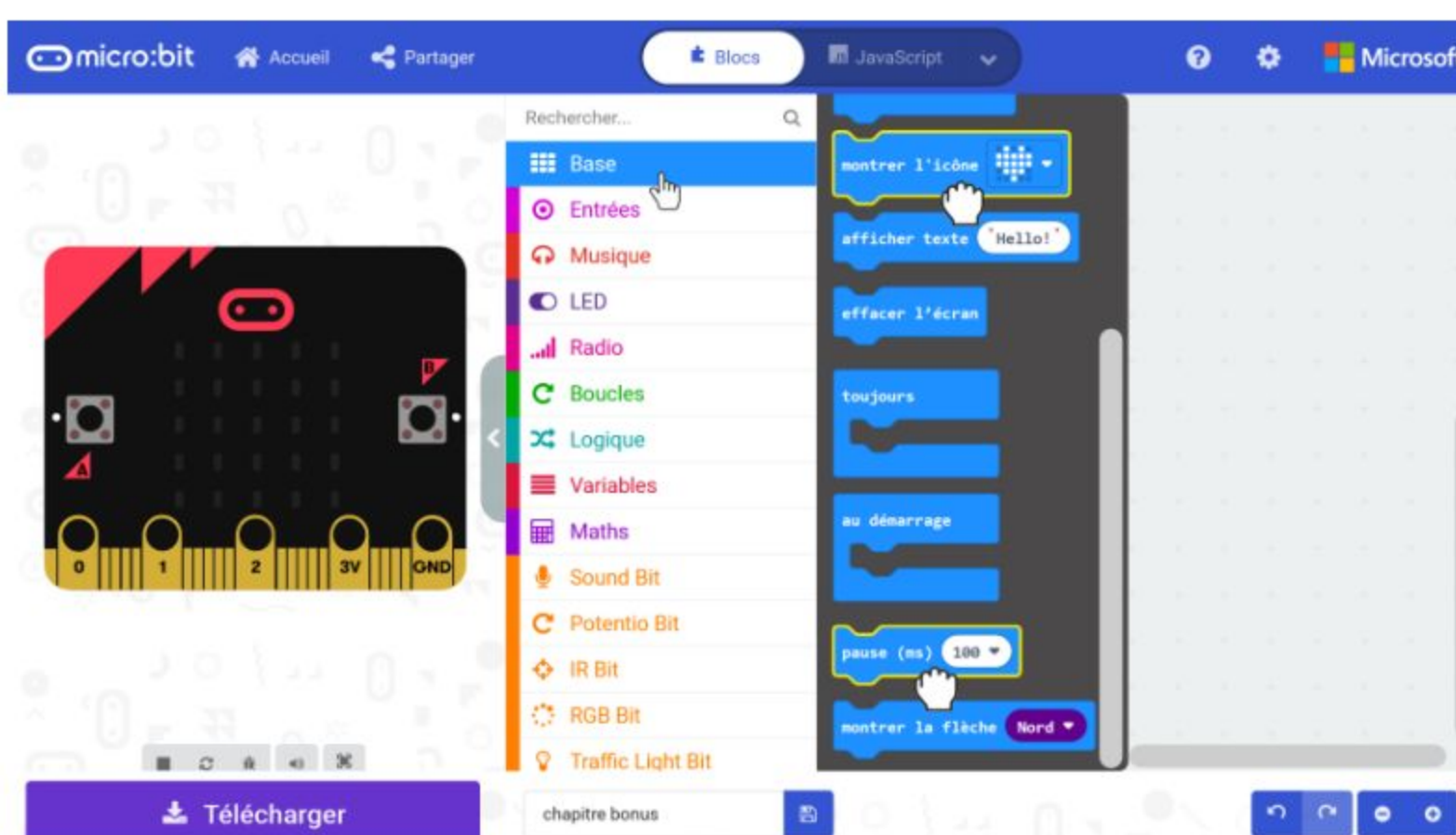


CHAPITRE BONUS : Jacques a dit, avec LEDs

Étape 13 Clique sur la catégorie [**Traffic Light Bit**] et sélectionne le bloc [**définir LED _ à _**]. Duplique-le et imbrique-les dans les trois premiers emplacements du bloc conditionnel. Change la couleur et l'état des LEDs comme indiqué ci-dessous.



Étape 14 Clique sur la catégorie [**Base**] et sélectionne le bloc [**pause _ (ms)**]. Duplique-le et intercale-les entre des blocs [**définir LED _ à _**] consécutifs et change la valeur en **500**. Ajoute le bloc [**Base**] : [**montrer l'icône**] dans le dernier emplacement du bloc conditionnel et change l'icône en "triste".





Étape 15 Clique sur la catégorie [**Musique**] et ajoute le bloc [**démarrer la mélodie _ répétition _**] pour compléter le code. Change la mélodie en “wawawawaa” (ou la mélodie de ton choix).

Voici le code complet :

Afin d'envoyer ou recevoir un signal radio, toutes les EDU:BITS en communication doivent être réglées sur le même groupe radio. Cela peut être n'importe quel nombre compris entre 0 et 255.

Ce sont des blocs événements. Lorsqu'ils sont déclenchés, EDU:BIT enverra la commande correspondante (ie : 1, 2, 3 ou 4) aux EDU:BITS à proximité réglées sur le même groupe radio.

Lorsqu'EDU:BIT reçoit une commande, elle vérifie la valeur et exécute le bloc du code associé.

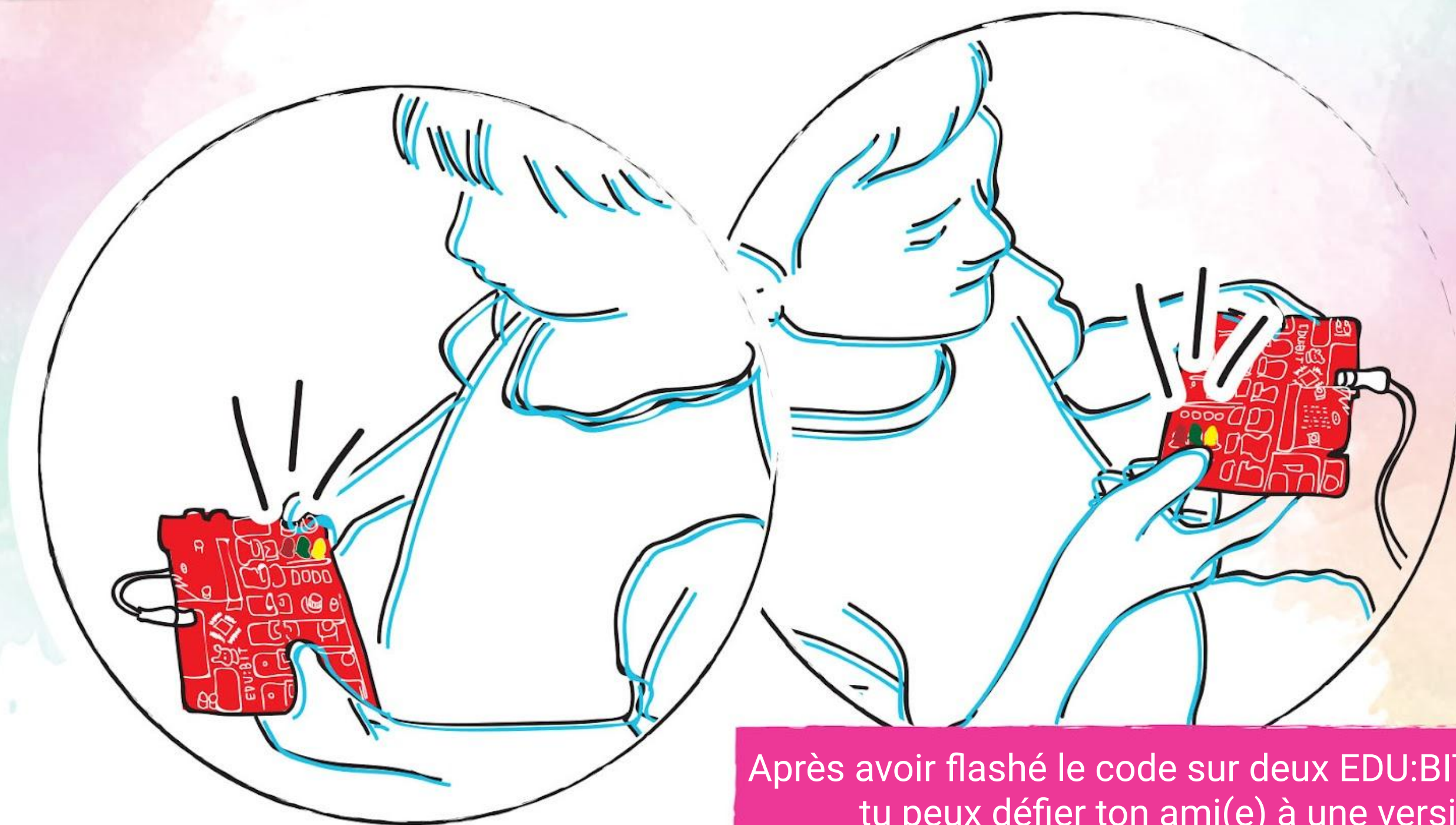
Étape 16 Flashe le code complet sur ton EDU:BIT et ceux de tes amis.



Lorsque deux EDU:BITS sont allumés, tu peux envoyer un signal radio pour allumer les LEDs des EDU:BITS de tes amis, et inversement, tes amis peuvent également appuyer sur les boutons de leur EDU:BIT pour allumer les LEDs de ta carte.

Jouons

Jacques a dit, avec LEDs



Après avoir flashé le code sur deux EDU:BITs, tu peux défier ton ami(e) à une version interactive du jeu Jacques a dit.

COMMENT JOUER :

Les deux joueurs incarnent Jacques à tour de rôle. Appuyer sur les boutons pour allumer les LEDs rouge, jaune et verte lorsque c'est ton tour.

— Pour démarrer le jeu, le joueur 1 allume une LED du Traffic Light Bit.

— Le joueur 2 observe et allume ensuite la même LED, suivie d'une autre LED.

— Le jeu continue, chaque joueur devant répéter la dernière séquence de LEDs allumées, puis en allume une supplémentaire qui vient s'y ajouter.

— Si un joueur allume la mauvaise LED (ou pas dans le bon ordre), alors l'adversaire incline son EDU:BIT pour mettre fin à la partie.

— Le perdant devra alors réinitialiser son EDU:BIT pour commencer une nouvelle partie.

Au début le jeu semble facile, mais il devient de plus en plus complexe à chaque tour de jeu. Pour gagner, tu dois observer la séquence attentivement. C'est un jeu amusant pour travailler sa mémoire.





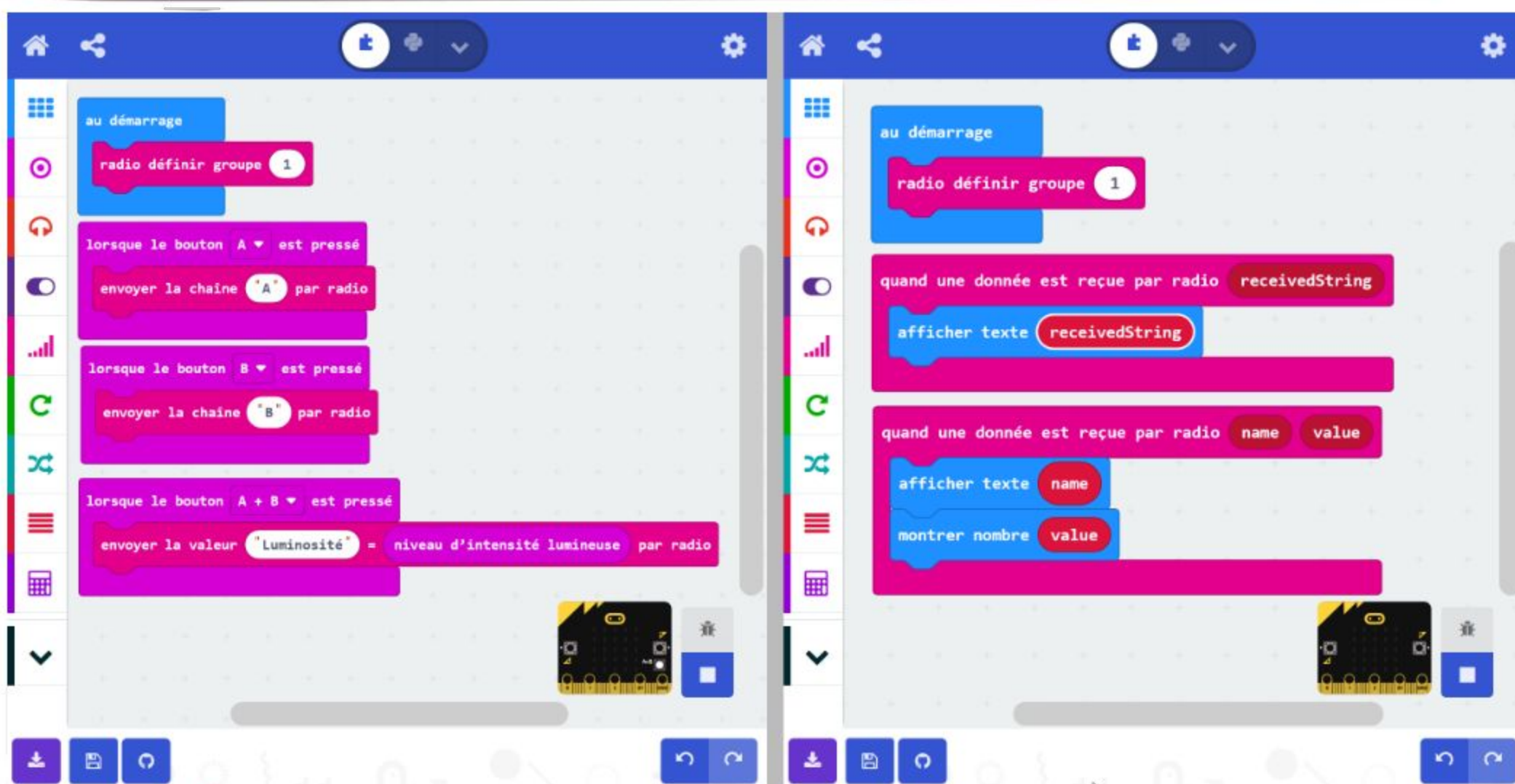
EXPLORE D'AUTRES BLOCS

#1 En plus d'envoyer des nombres, tu peux envoyer des chaînes de caractères en utilisant le bloc [**envoyer la chaîne _ par radio**]. Tu devras utiliser le bloc [**quand une donnée est reçue par radio receivedString**] pour recevoir le signal de diffusion du message. La longueur maximale de la chaîne est de 19 caractères.

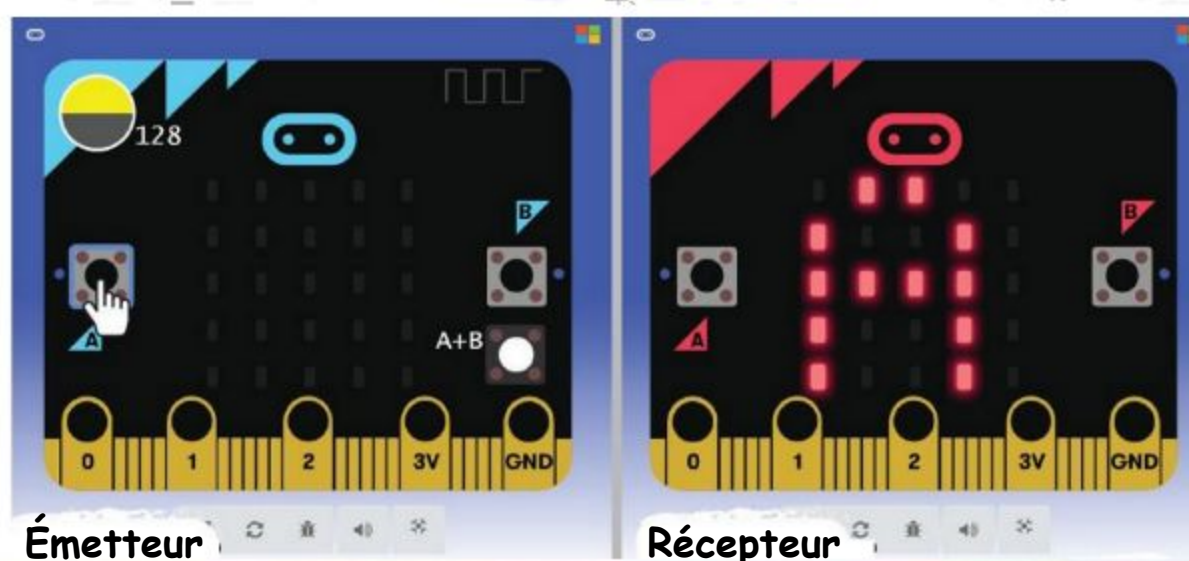
#2 Utilise les blocs [**envoyer la valeur "_" = _**] et [**quand une donnée est reçue par radio name value**] si tu as besoin d'envoyer une chaîne et un nombre ensemble. La longueur maximale de la chaîne est de 8 caractères.



Si tu n'as pas accès à plusieurs EDU:BITs, tu peux toujours tester les fonctions de communication radio. Rends-toi sur makecode.com/multi pour écrire ton code pour "l'émetteur" et "le récepteur". Tu peux voir le résultat sur le simulateur.



Clique sur la carte micro:bit pour l'ouvrir en plein écran.



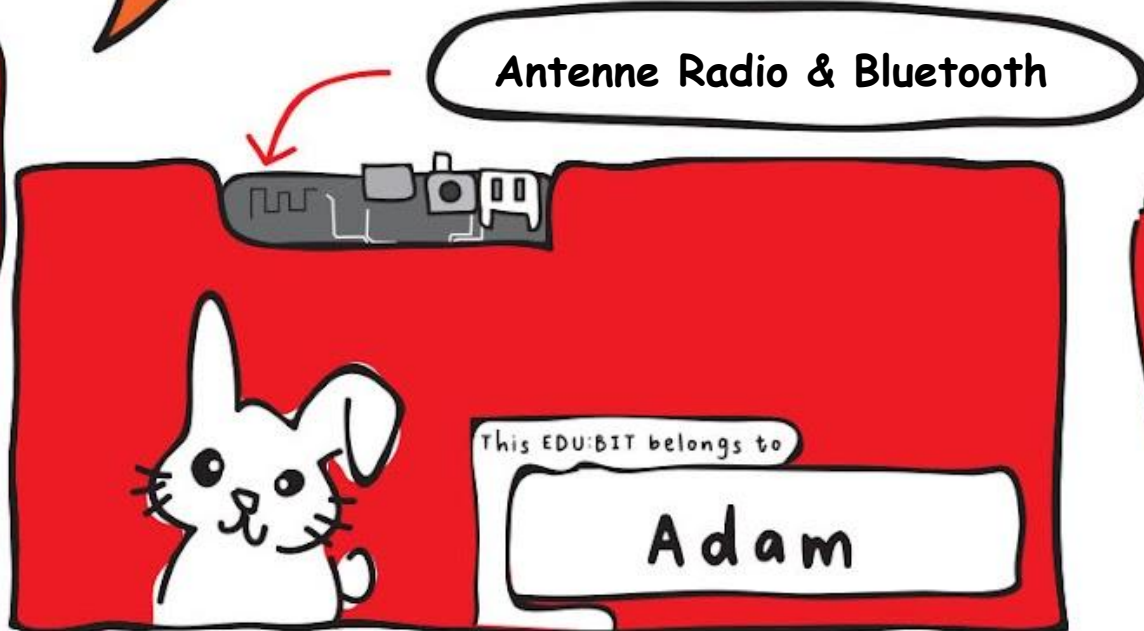
Lorsque tu appuies sur le bouton A du micro:bit émetteur, le micro:bit récepteur reçoit le signal radio et affiche la chaîne reçue, ie : "A". Que se passe-t-il si tu appuies sur A+B ?



FAIT AMUSANT!



T'as vu ! J'ai pu me souvenir de la séquence. À ton tour maintenant, Adam.



Antenne Radio & Bluetooth

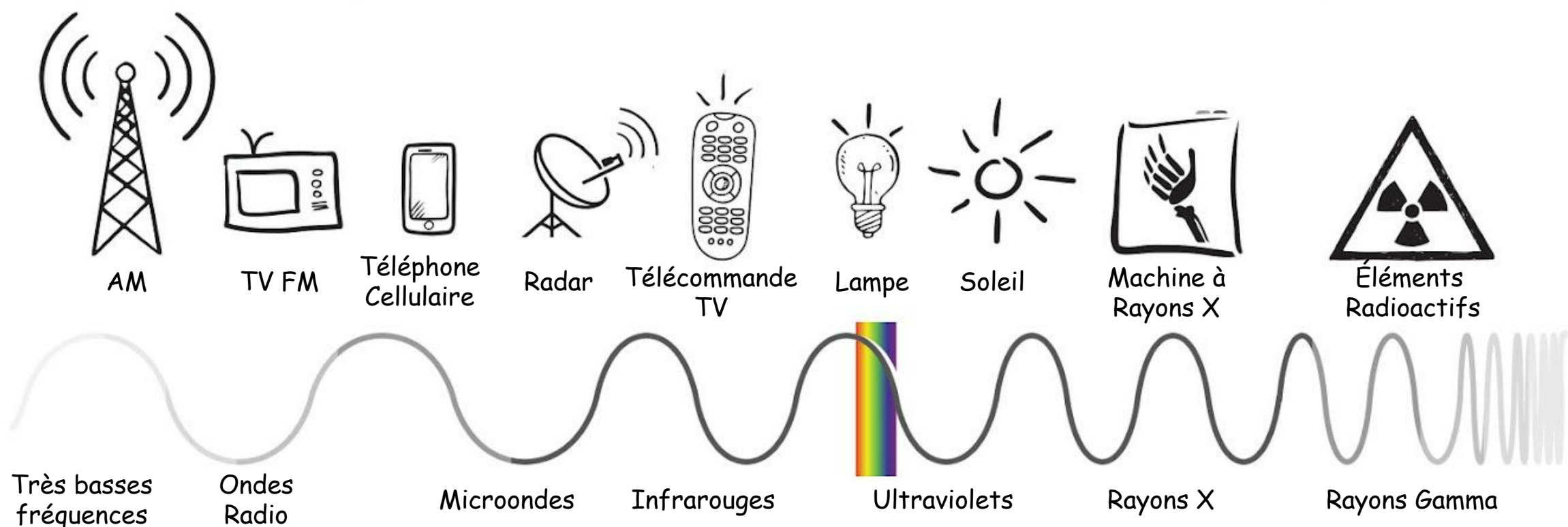


Je l'ai... et maintenant, c'est de nouveau à toi. Voyons si tu peux retenir ça !

Si tu tournes EDU:BIT vers le bas, tu verras l'antenne radio et Bluetooth intégrée ici.

L'antenne transmet des signaux sous forme d'ondes radio électromagnétiques, qui sont couramment utilisées pour la télévision et les émissions radio, ou encore des transmissions par satellite.

Spectres Électromagnétiques



NOTE!

Pour que ton EDU:BIT puisse envoyer ou recevoir un signal radio à d'autres / depuis d'autres EDU:BIT, tu dois toutes les régler sur le même groupe radio.



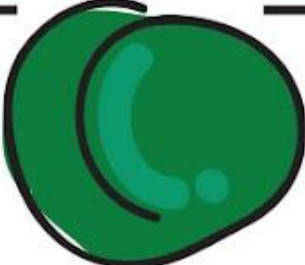
APPLICATION DÉFI

Définis un système d'évaluation par communication réseau pour ta classe.

Comment ça marche ?

Règle tous les EDU:BIT de la classe sur le même groupe radio.

L'EDU:BIT de l'enseignant est configuré pour afficher un texte lorsqu'il reçoit une chaîne de caractères, et allume une LED du Traffic Light Bit lorsqu'il reçoit un nombre par signal radio, tels que ...

Nombre reçu	LED allumée	Quelle signification ?
1	ROUGE 	A / Non / Faux
2	JAUNE 	B / Peut-être / Pas sûr
3	VERTE 	C / Oui / Vrai

Les EDU:BITs des élèves envoient une chaîne de caractères contenant leur nom et un numéro (1, 2 ou 3) pour allumer une des LEDs de l'EDU:BIT de l'enseignant.

Appuie sur le bouton A pour envoyer le numéro 1.

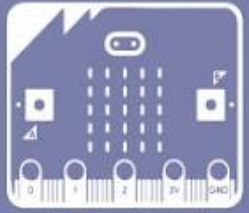
Appuie sur le bouton B pour envoyer le numéro 2.

Appuie sur les boutons A+B pour envoyer le numéro 3.

Voici un conseil pour toi. Attribue des surnoms courts (avec seulement 2 ou 3 caractères) pour chaque élève afin de réduire le temps d'affichage.



J'ai appris à...



- Programmer EDU:BIT pour afficher un texte et une animation sur la matrice de LEDs.
- Télécharger, sauvegarder, publier et éditer un fichier MakeCode .hex.



- Utiliser des blocs d'entrées pour la programmation événementielle.
- Créer et utiliser des variables.



- Utiliser un buzzer piézoélectrique du Music Bit pour jouer des mélodies simples.
- Créer et utiliser des fonctions.
- Lire des partitions de musique.



- Programmer EDU:BIT pour contrôler des LEDs du Traffic Light Bit - *allumer, éteindre & basculer.*
- Ajouter des extensions à l'éditeur MakeCode.



- Programmer EDU:BIT pour détecter des objets grâce au capteur infrarouge IR BIT.
- Utiliser une boucle tant que.
- Créer et utiliser des tableaux.



- Programmer EDU:BIT pour lire des valeurs analogiques provenant du Potentio Bit.
- Mapper des valeurs analogiques lues.
- Utiliser des blocs logiques pour la programmation conditionnelle.



- Programmer EDU:BIT pour détecter du bruit avec le Sound Bit.
- Utiliser des événements à déclencher pour naviguer entre différents modes.



- Programmer EDU:BIT pour contrôler un moteur DC - contrôle du sens et de la vitesse de rotation.
- Utiliser des blocs mathématiques pour réaliser des opérations arithmétiques.



- Programmer EDU:BIT pour contrôler un servomoteur - position angulaire.



- Programmer EDU:BIT pour allumer des LEDs RVB sur le RGB Bit, afficher différentes couleurs et différents motifs.



- Programmer EDU:BIT pour envoyer et recevoir des messages par signal radio.

* Coche la case si tu maîtrises la compétence ; sinon, revient sur le chapitre pour réviser.

NOTE DE RERO EDUTEAM @ CYTRON

FÉLICITATIONS !!!

Tu as réussi tous les chapitres et appris à coder avec l'éditeur MakeCode. Nous espérons que tu as passé de bons moments et as joué à notre sélection de jeux pour enfants populaires. Et pouce en l'air pour avoir développé quelques applications pratiques pour ta classe.

Maintenant, tu dois avoir une bonne idée de ce qu'il est possible de faire avec une carte micro:bit et tous ses modules complémentaires (capteurs et actionneurs) comme ceux de ta carte EDU:BIT. Psst... savais-tu que tu peux casser la carte et séparer chacun des modules ?

Vas-y, "détache-les" si tu veux. Une fois séparés de la carte principale, tu peux créer de nouveaux projets avec les différents modules. Tu devras les connecter à l'aide des câbles plug-and-play fournis.

Et maintenant, il est temps pour toi de te creuser la tête et faire travailler tes méninges pour développer de nouveaux jeux et de nouvelles applications. Nous avons hâte de voir quels fabuleux projets tu vas élaborer.

Pense à partager tes créations avec nous. Envoie-nous un email ou laisse-nous un message sur notre page FaceBook. Nous aimerions avoir de tes nouvelles.

Bien sincèrement,
Adam & Anna



Informe-nous
de tes progrès !



[link.cytron.io/
edubit-resource-hub](https://link.cytron.io/edubit-resource-hub)



