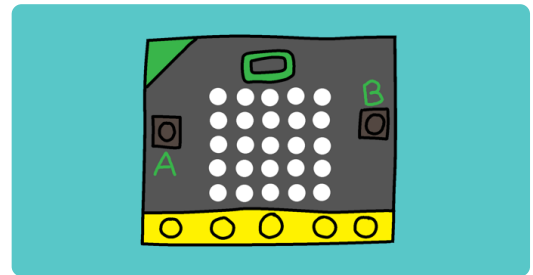


 Projets

Frustration

Guider une baguette le long d'un circuit sans le toucher.



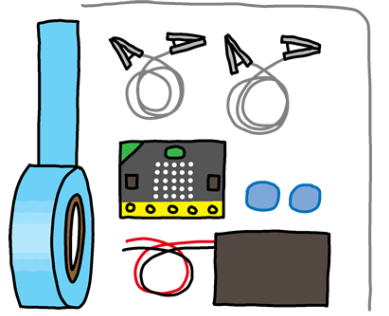
Étape 1 Introduction

Tu vas faire un jeu dans lequel les joueurs devront guider un baguette le long d'un circuit sans le toucher. Toucher ajoutera un au score du joueur, le joueur avec le plus petit score gagne !

Instructions : Si tu lis ceci en ligne , appuie sur **A** sur le micro:bit ci-dessous pour démarrer le jeu, puis touche la broche **0** pour faire une connexion.

Pour ce projet, tu auras besoin de quelques éléments supplémentaires :

- Fil de fer (environ 50 cm) ;
- Pâte à modeler (Plasticine ou similaire, doit être non conductible) ;
- Ruban adhésif isolant (optionnel) ;
- Pincettes crocodile (optionnel).



Informations complémentaires pour les responsables de club

Si vous avez besoin d'imprimer ce projet, merci d'utiliser la **version imprimable** (<https://projects.raspberrypi.org/fr-FR/projects/frustration/print>).



Notes pour le responsable de club

Introduction :

Dans ce projet, les enfants apprendront à utiliser des variables en faisant un jeu où il faut suivre le fil. Le but du jeu est de guider un baguette le long d'un circuit sans le toucher. Toucher ajoutera un au score du joueur, le joueur avec le plus petit score gagne.

Ressources

Pour mener à bien ce projet, les enfants auront besoin :

- Fil de fer (environ 50 cm) ;
- Pâte à modeler ;
- Ruban adhésif isolant (optionnel) ;
- Pincettes crocodile (optionnel).

Les étapes 1 et 2 de ce projet peuvent être réalisées sans cet équipement.

Pour ce projet, l'éditeur **MakeCode (PXT)** (<http://jumpto.cc/pxt-new>) devrait être utilisé.

Vous pouvez trouver une version terminée de ce projet sur makecode.microbit.org/#pub:71088-38400-20107-39742 (<https://makecode.microbit.org/#pub:71088-38400-20107-39742>), et le fichier compilé .hex peut être téléchargé en cliquant sur le lien « Télécharger le matériel du projet » pour ce projet, qui contient :

- Frustration.hex

Objectifs d'apprentissage

- Variables ;
- Broches d'entrée / sortie (E / S).

Ce projet couvre les éléments des sections suivantes du **Raspberry Pi Digital Making Curriculum** (<http://rpf.io/curriculum>) :

- **Utiliser des concepts basiques de programmation pour créer des programmes simples.** (<https://www.raspberrypi.org/curriculum/programming/creator>)
- **Utiliser des composants basiques numériques, analogiques, ou électromécaniques.** (<https://www.raspberrypi.org/curriculum/physical-computing/creator>)

Défis

- « Afficher une image » - Afficher une images sur le micro:bit avant que le nombre d'**échecs** ne soit affiché.
- « Mode triche ! » - Soustrait 1 du nombre d'**échecs** lorsque le bouton B est pressé.
- « Personnalise ton jeu » - Fais des changements dans la baguette et le circuit pour rendre le jeu plus difficile.



Matériel pour projet

Ressources pour le responsable de club

- projet terminé en ligne (<https://makecode.microbit.org/#pub:71088-38400-20107-39742>)
- Fichier projet .hex à transférer sur ton micro:bit (<https://projects-static.raspberrypi.org/projects/frustration/10911ce3bba8ebdb61f6e5d00a6d81074e962a13/fr-FR/resources/micro-bit-Frustration.hex>).

Étape 2 Stocker les échecs

Commençons par créer un endroit où stocker le nombre d'échecs.

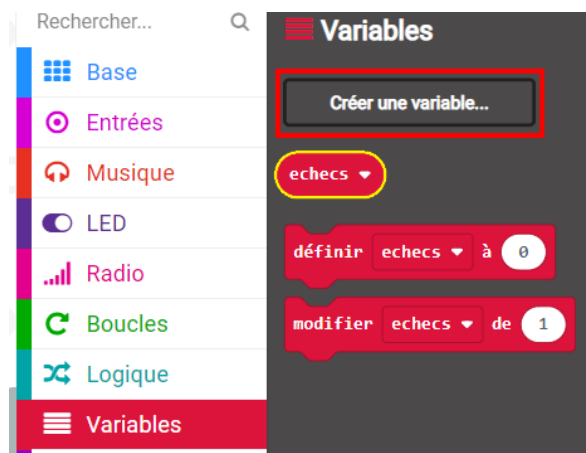
- Va sur rpf.io/microbit-new (<https://rpf.io/microbit-new>) pour démarrer un nouveau projet dans l'éditeur MakeCode (PXT). Appelle ton nouveau projet « Frustration ».
- Supprime les blocs **toujours** et **au démarrage** en les faisant glisser vers la palette :



- Une nouvelle partie commence quand le joueur appuie sur le bouton A. Clique sur « Entrée », puis **Lorsque le bouton A est pressé**.



- Maintenant tu as besoin d'une variable pour stocker le nombre de fois où tu vas échouer en touchant le fil avec la baguette. Clique sur « Variables », puis « Créer une variable ». Nomme la variable **echecs**.

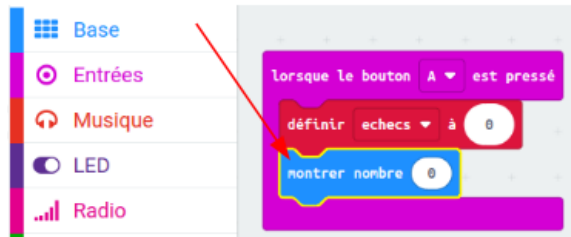


- Place un bloc **définir ... à** depuis « Variables » et sélectionne **echecs** :

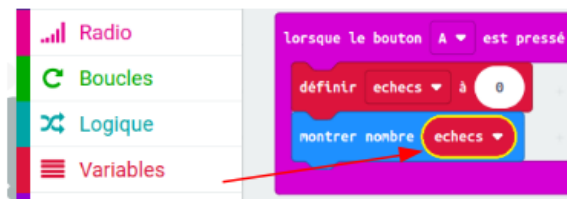


Ceci mettra à zéro le nombre d'échecs quand on appuie sur le bouton A.

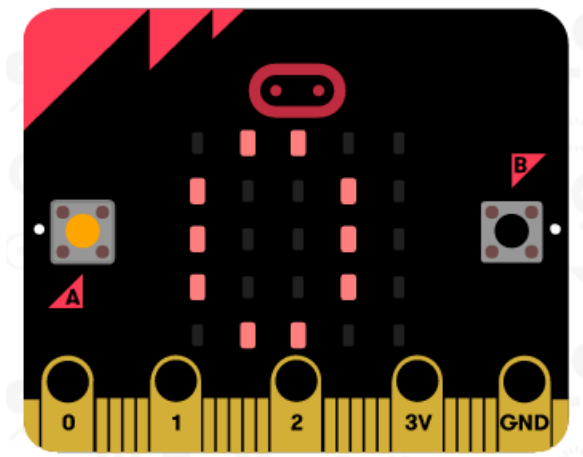
- Enfin, tu peux afficher le nombre d'échecs sur ton micro:bit. Pour cela, place un bloc `montrer nombre`, depuis « Base », à la fin du script.



- Ensuite place `echecs` depuis « Variables » dans le bloc `montrer nombre`.

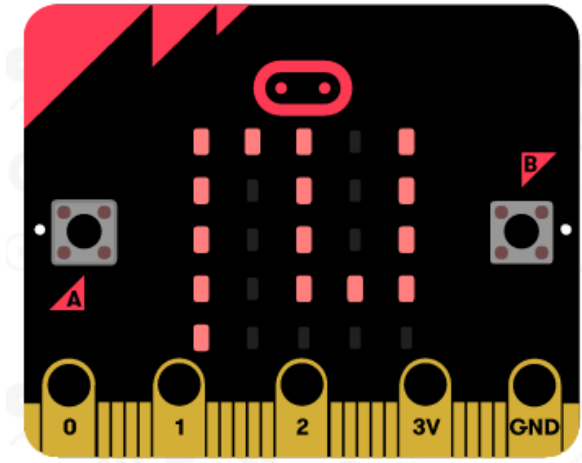


- Clique sur « lancer » pour tester ton script. Cliquer sur le bouton A devrait afficher le nombre d'échecs, qui a été mis à 0.

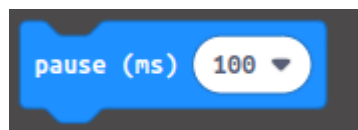


Étape 3 Défi : Afficher une image

Sauras-tu afficher une image pendant 1 seconde (1000 ms) avant que le nombre d'échecs soit affiché ?



Tu devras utiliser les blocs suivants depuis Base pour atteindre ton but :



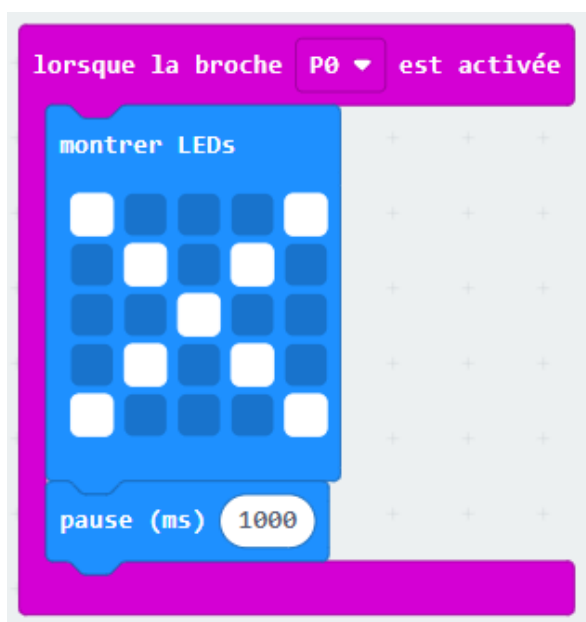
Étape 4 Garder la trace des échecs

Ajoutons du code pour garder une trace des échecs.

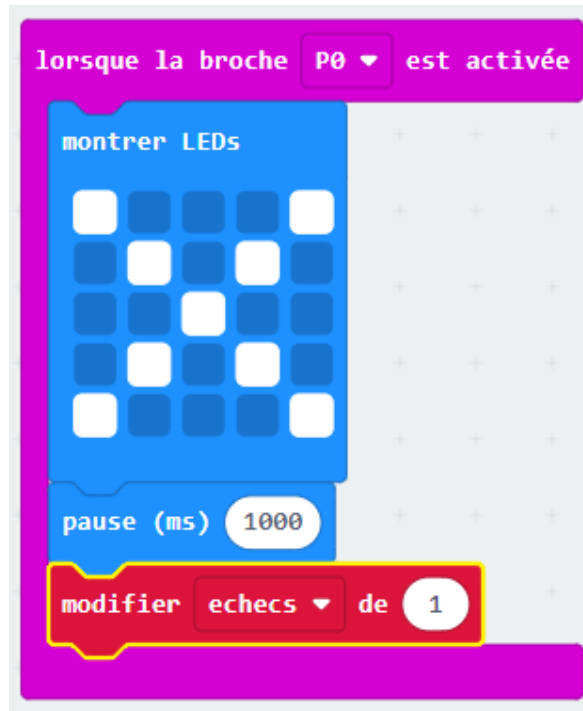
- Tu vas ajouter 1 à la variable `echec` chaque fois qu'une connexion est faite sur la broche P0. Pour cela, prend un bloc `lorsque la broche P0 est activée` dans « Entrées ».



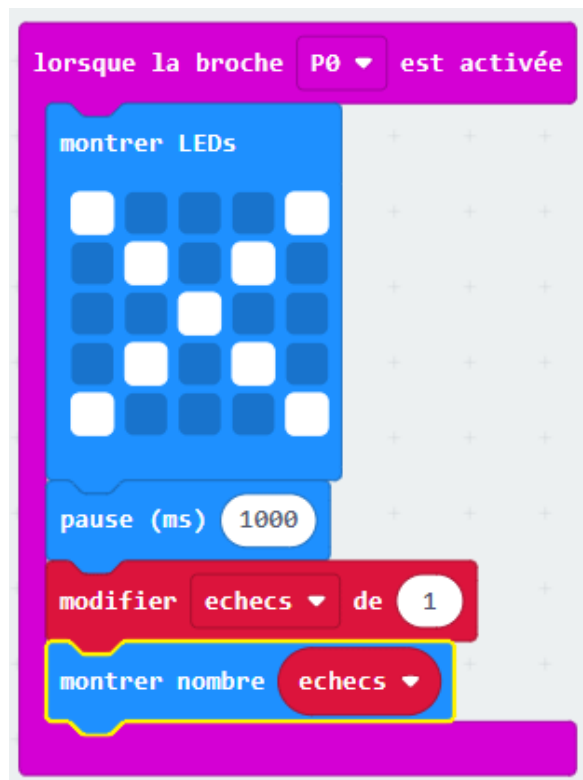
- Ensuite, ajoute 2 blocs pour afficher une croix pendant 1 seconde lorsque la broche P0 est activée.



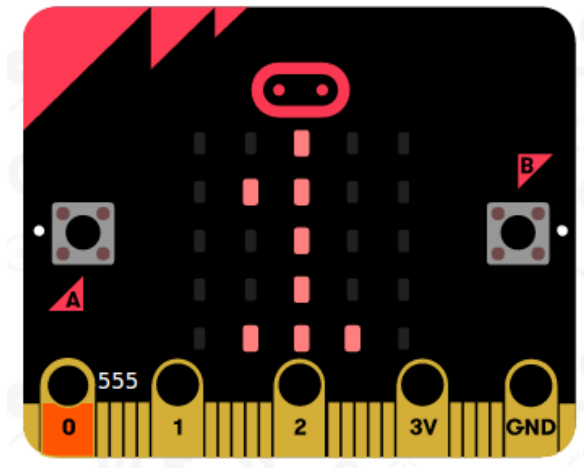
- Tu devras ensuite ajouter 1 à la variable `echecs`. Pour cela, prend un bloc `modifier objet de 1` dans « Variables » et remplace `objet` par `echecs`.



- Enfin, tu peux ajouter du code pour afficher le nombre d'échecs mis à jour. Voici à quoi devrait ressembler ton code.



- Teste ton code en appuyant sur le bouton A de l'émulateur pour démarrer ton jeu. Chaque fois que tu appuies sur la broche P0, tu devrais voir la variable `echecs` augmenter de 1.

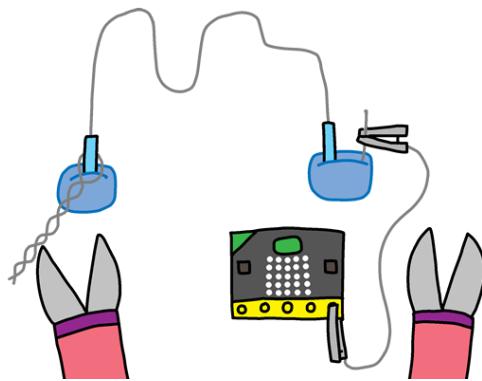


- Clique sur « Télécharger » et transfère ton script sur ton micro:bit. Tu peux appuyer sur la broche P0 en fermant le circuit. Pour cela, place ton pouce droit sur la masse (broche GND) et touche la broche P0 avec ton pouce gauche.

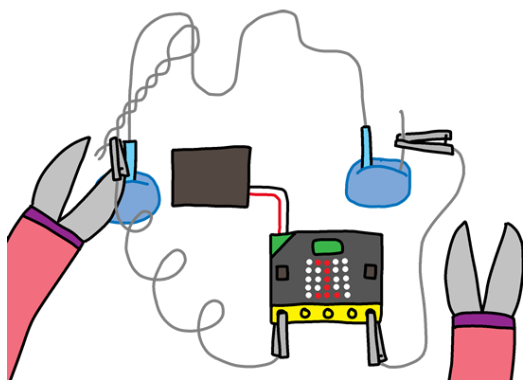
Étape 5 Construire ton jeu

Maintenant que tu as programmé ton jeu, assemblons tout ensemble !

- D'abord, créons une baguette. Prends un morceau de fil de fer d'environ 20 cm de long et replie-le en deux, en créant un boucle à l'extrémité.
- Tu peux ensuite entortiller les 2 extrémités du fil ensemble.
- Pour faire le circuit, prends un autre morceau de fil d'environ 30 cm de long et tords la partie centrale du fil pour lui donner une forme biscornue. Tu devras tordre vers le haut une des extrémités du circuit.
- Si tu as du ruban adhésif isolant, enveloppe les 2 extrémités du circuit, en laissant dépasser un peu de métal à chacune des extrémités.
- Enfile ta baguette sur le circuit, et pique les extrémités du fil dans la pâte à modeler pour que le circuit tienne debout tout seul.
- Tu peux brancher ton jeu à ton micro:bit en utilisant des pinces crocodiles ou du fil. Connecte d'abord la masse (broche GND) à l'une des extrémités du circuit.



- Tu peux ensuite connecter la broche P0 à ta baguette.
- Teste ton jeu. Appuie sur le bouton A et ton score sera remis à 0. Chaque fois que ta baguette touche le circuit, le courant passe entre la baguette et le circuit et ton micro:bit ajoute 1 au nombre d'échecs.



Étape 6 Défi : Mode triche !

Sauras-tu ajouter un peu de triche dans ton jeu, et faire en sorte qu'appuyer sur le bouton B diminue ton score de 1 ?

Étape 7 Défi : Personnalise ton jeu

Demande à tes amis d'essayer ton jeu. Si ton jeu est trop facile, tu peux le rendre plus difficile en :

- Créant un circuit plus long ;
 - En ajoutant plus de virages dans ton circuit ;
 - En réduisant la taille de la boucle de ta baguette.
-

Ce projet a été traduit par des bénévoles:

Michel Arnols

Jonathan Vannieuwerkerke

Grâce aux bénévoles, nous pouvons donner aux gens du monde entier la chance d'apprendre dans leur propre langue. Vous pouvez nous aider à atteindre plus de personnes en vous portant volontaire pour la traduction - plus d'informations sur rpf.io/translate (<https://rpf.io/translate>).

Publié par **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) sous un **Creative Commons license** (<https://creativecommons.org/licenses/by-sa/4.0/>).

Voir le projet et la licence sur GitHub (<https://github.com/RaspberryPiLearning/frustration>).