
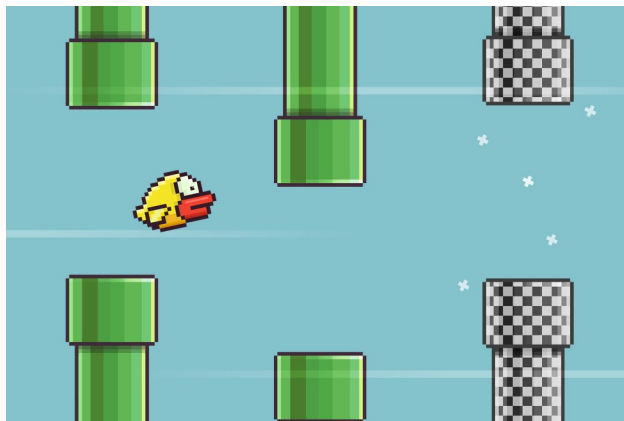


Design, Innovation et Créativité (DIC)	Objets Techniques, les Services et les Changements Induits dans la Société (OTSCIS)	La Modélisation et la Simulation des Objets et Systèmes Techniques (MSOST)	L'Informatique et la Programmation (IP)
	<b>Flappy Bird</b>		<b>Cycle 4 - 3eme</b>
			Séquence 2 -Activité 1
<u>Je vais apprendre ce qu'est :</u> ⇒ IP 2.3.3 Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles. ⇒ IP 2.3.2 Notion de variable informatique. ⇒ IP 2.3.1 Notions d'algorithme et de programme. ⇒ IP 2.2 Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.		<u>Socle commun :</u> → CT 5.4 Piloter un système connecté localement ou à distance. → CT 5.5 Modifier ou paramétrer le fonctionnement d'un objet communicant.	



## Comment réaliser un jeu vidéo simple sur une carte programmable ?

Il s'agit d'une version simplifiée du jeu Flappy Bird. L'objectif est de diriger un oiseau, qui se déplace continuellement vers le haut ou le bas, pour éviter les obstacles. Si le joueur touche un obstacle, il perd.

Le but de ce didacticiel est d'enseigner les bases des sprites, des tableaux et des boucles de jeu.

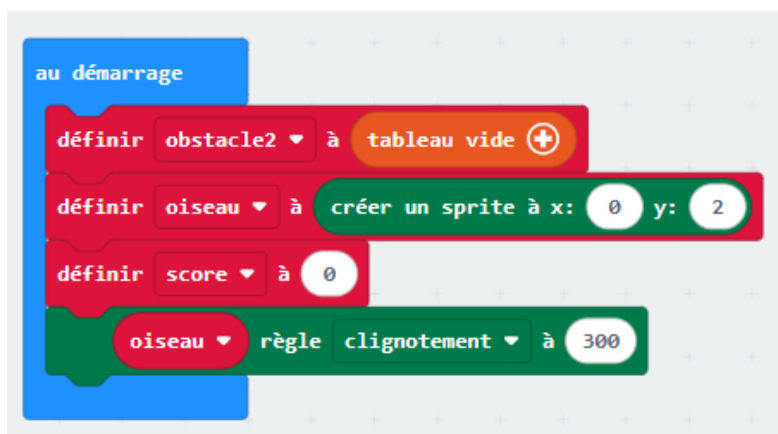
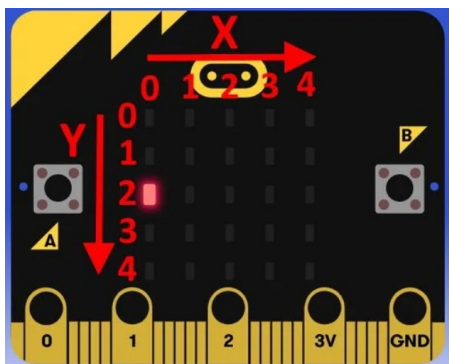
### → Étape 1 : Ajouter l'oiseau au jeu et déclarer les variables

Tout d'abord, nous allons ajouter un sprite pour l'oiseau depuis le menu Jeu et le faire clignoter.

Il faudra créer la variable "bird". Le mot bird veut dire oiseau.

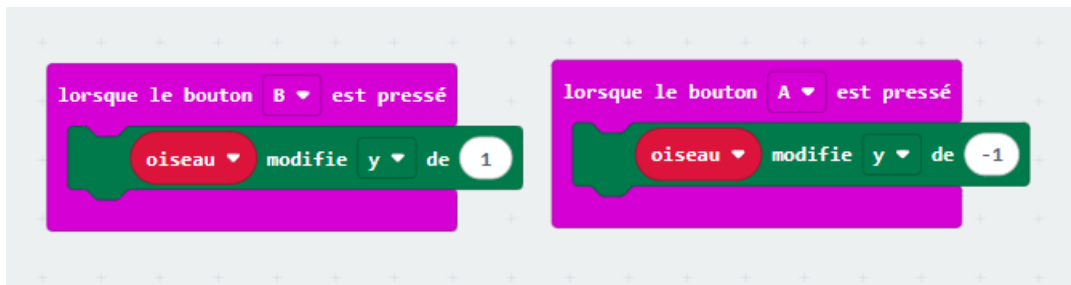
Le mot sprite veut dire lutin.

x correspond à l'axe des abscisses et y à l'axe des ordonnées.



## → Étape 2 : Faire voler l'oiseau

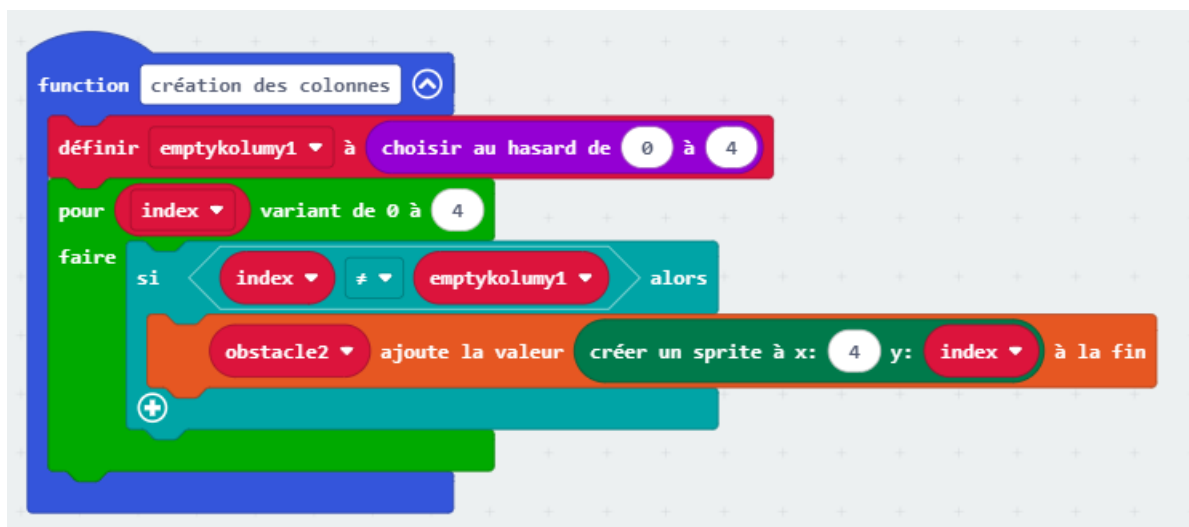
Avant de créer le code des actions du jeu, ajoutons d'abord quelques contrôles afin de pouvoir nous déplacer. Nous allons contrôler l'oiseau en appuyant sur le bouton A pour monter ou sur le bouton B pour descendre.



## → Étape 3 : Générer des colonnes

C'est là que les choses commenceront à devenir intéressantes. Nous allons générer aléatoirement des obstacles. Tous les obstacles auront un seul trou pour que l'oiseau puisse voler à travers.

Créer la fonction « création des colonnes » :



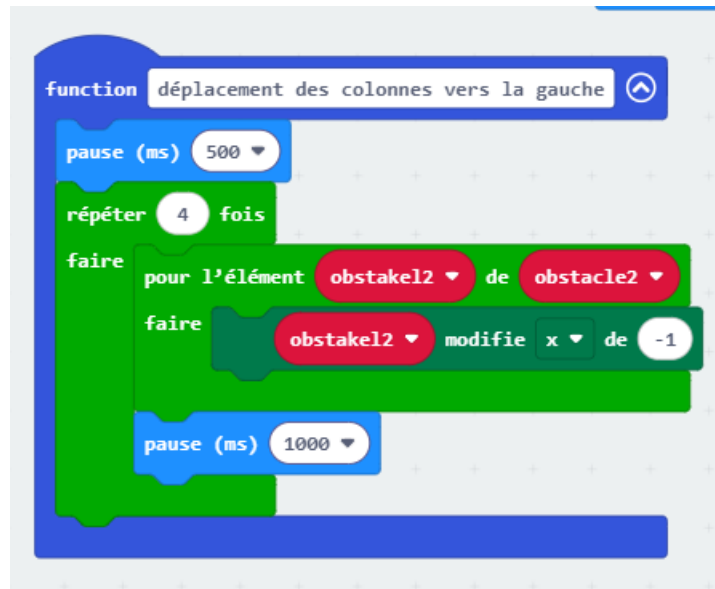
Maintenant, à chaque redémarrage de micro:bit, vous devriez voir différents obstacles verticaux générés automatiquement.

Avant de continuer, assurez-vous que les obstacles sont générés de manière aléatoire et que l'oiseau monte et descend.

## → Étape 4 : Faire bouger les colonnes

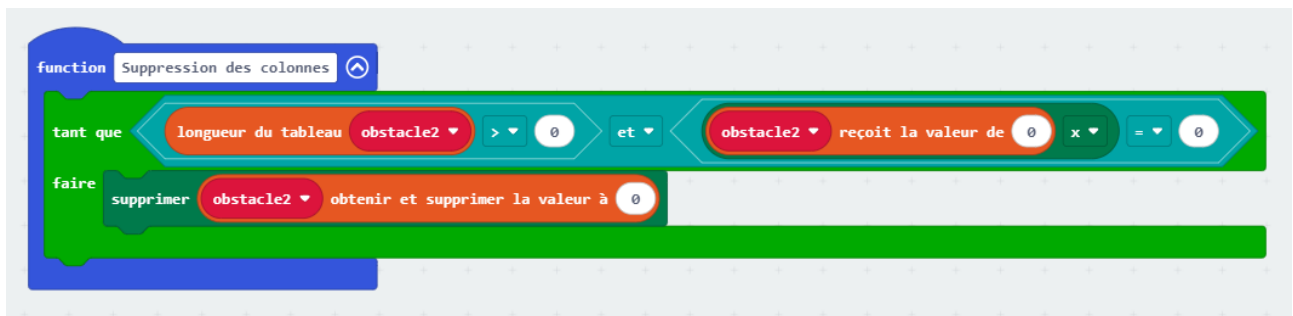
Créer la fonction « déplacement des colonnes vers la gauche ».

Les obstacles doivent se déplacer vers la gauche toutes les secondes.



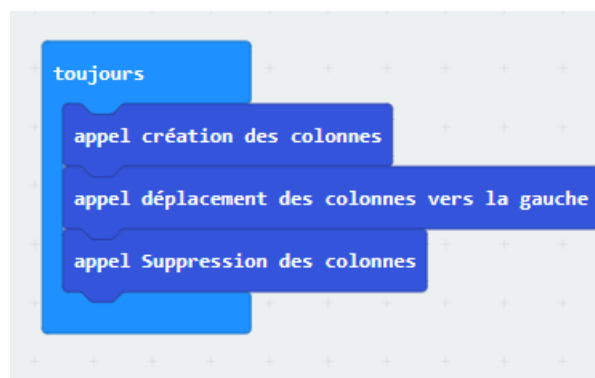
## → Étape 5 : Faites disparaître les colonnes

Faites disparaître les obstacles après avoir atteint le coin le plus à gauche.



## → Étape 6 : Appeler les 3 fonctions créées

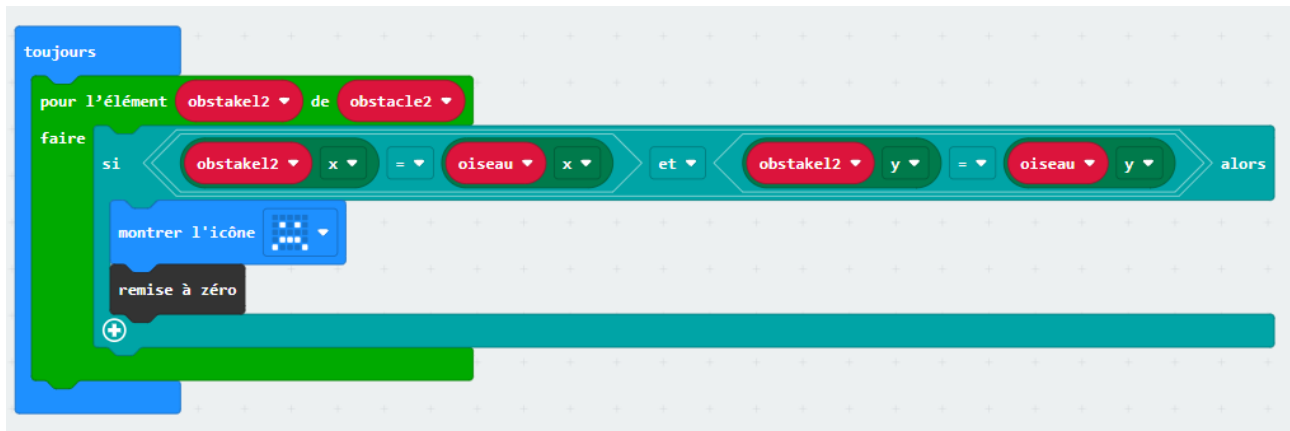
Dans un bloc « Toujours », appelez les 3 fonctions précédemment créées.



## → Étape 7 : Fin du jeu

À l'heure actuelle, rien ne se passe lorsque l'oiseau est heurté par un obstacle.

Corrigez cela :



→ Testez votre jeu

→ Appelez Mme FENEUX pour vérification

→ Téléverser votre jeu dans votre carte Micro:bit

→ Testez votre jeu sur la carte Micro:bit