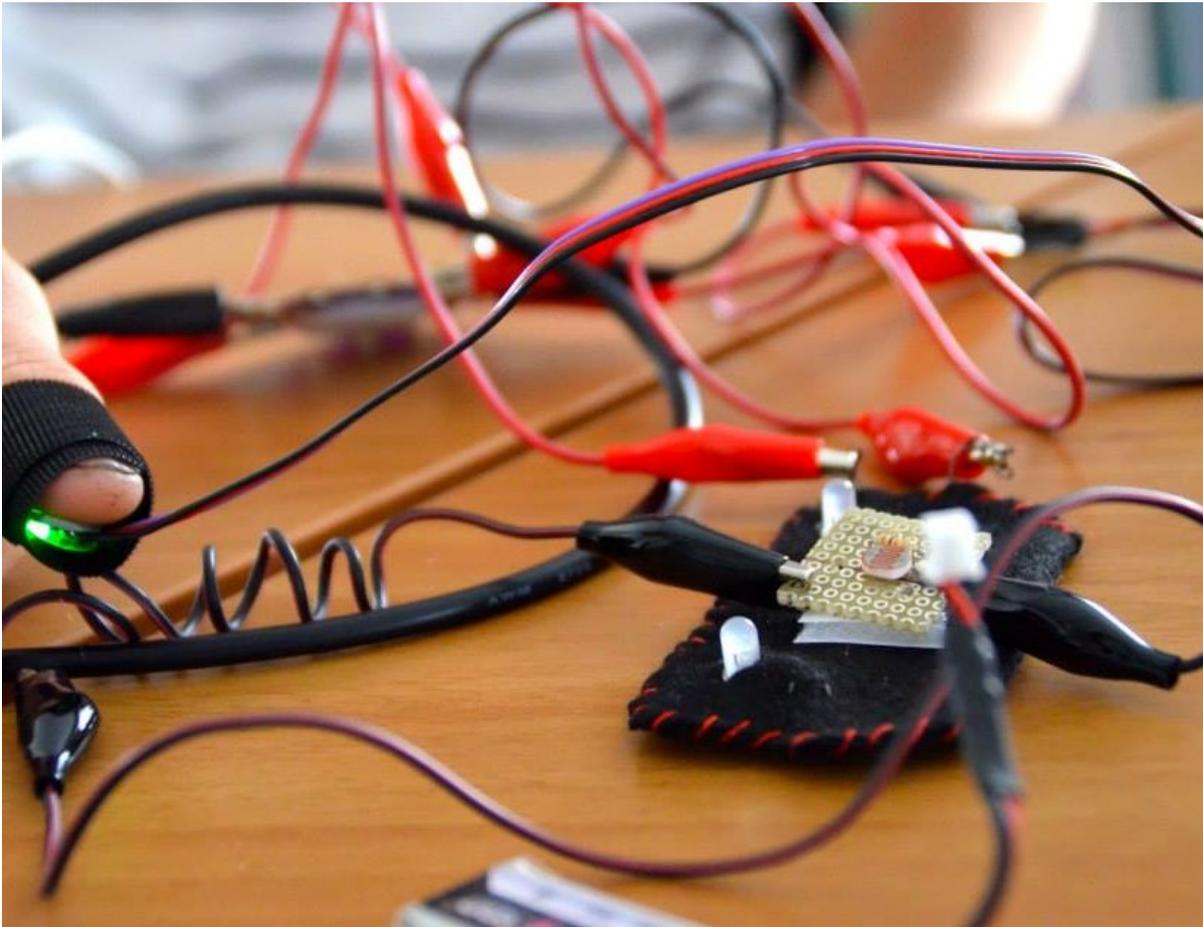


## L'IOT à des fins éducatives et scolaires

La tâche de ce manuel est d'introduire différentes méthodologies, scénarios d'apprentissage et activités éducatives sur le codage et l'IoT dans l'éducation.





Pour prospérer, ils doivent apprendre à concevoir des solutions innovantes aux problèmes inattendus qui se poseront sans aucun doute dans leur vie. Leur succès et leur satisfaction seront basés sur leur capacité à penser et à agir de manière créative. Les connaissances seules ne suffisent pas: elles doivent apprendre à les utiliser de manière créative

- Mitchel Resnick, MIT Media Lab

# Une introduction à l'Internet des objets et des appareils pouvant être portés, dans l'éducation

Dans ce manuel, nous allons apprendre à utiliser certains éditeurs en ligne pour programmer et interagir avec des capteurs et des sorties simples. L'Internet des objets est un domaine croissant du marché, des thermostats à la montre intelligente.

Dans le domaine de l'éducation, il est important d'introduire toute cette technologie car pour l'étudiant, c'est une activité engageante qui permet d'appliquer la théorie des 4P.

Project, Peer, Play and Passion, (Projet, partage, jeu et passion), une méthodologie présentée par Mitchel Resnick du MIT Lifelong Learning Lab (MediaLab) et qui convient très bien aux activités éducatives de l'IoT. De plus, dans ce manuel, nous présentons la possibilité de créer un programme avec des capteurs présents dans différentes plates-formes commerciales, tous liés à l'éditeur en ligne Scratch 3. Dans les scénarios d'apprentissage, nous utiliserons également différents programmes comme Snap pour Arduino ou makecode. Tous les logiciels utilisés sont gratuits et compatibles avec les plateformes robotiques les plus importantes comme Lego, Microbit, Arduino et Raspberry PI.

## Introduction à Arduino et assimilés

Nous avons choisi de ne présenter que l'Arduino et les plate-formes compatibles car ils sont open source et, dans le cadre d'un projet européen, nous considérons éthique l'utilisation de plates-formes open source bon marché comme Arduino et Elegoo.

Ces shields sont totalement compatibles avec les capteurs et les sorties poletora os.

Arduino est la plate-forme électronique open source la plus populaire qui change le monde de l'éducation, grâce au bas prix et à la facilité d'utilisation du prototypage et de la programmation.

Un Shield typique est composé d'un microcontrôleur 8 bits avec des puces différentes de la famille Mega AVR. Pour chaque Shield, il y a une entrée numérique, une sortie numérique et une entrée et une sortie analogiques.

## Logiciel gratuit en ligne

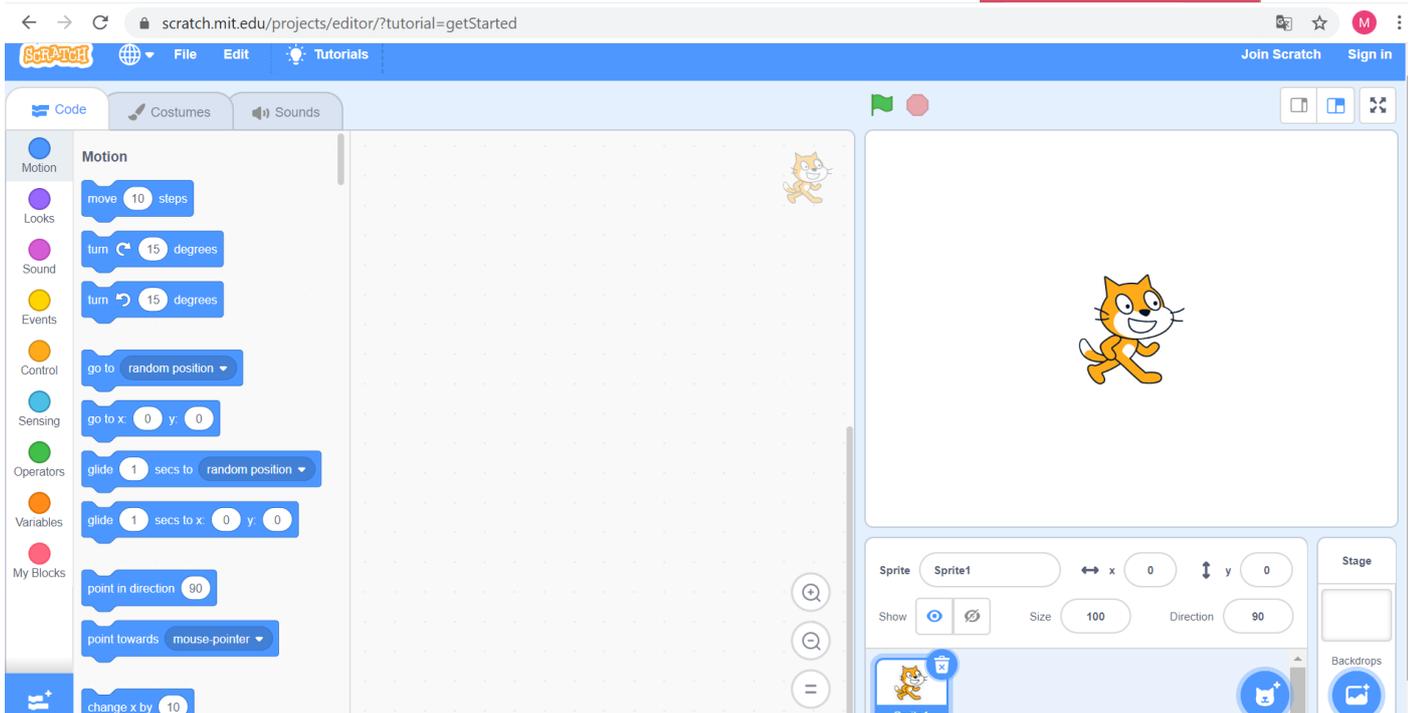
Dans ce projet, nous choisissons d'utiliser uniquement des logiciels gratuits pour programmer nos objets. Le programme le plus populaire est Scratch, maintenant dans sa version Scratch3.

### Scratch

Scratch est un éditeur de programmation visuelle développé par MIT Media Lab. Scratch est né en 2006 et est maintenant utilisé dans la plupart des écoles du monde. Il est disponible dans plus de 70 langues.

Scratch 2 est également disponible en mode offline. Vous pouvez le télécharger ici <https://scratch.mit.edu/download>. Scratch 2 n'est pas disponible sur les tablettes.

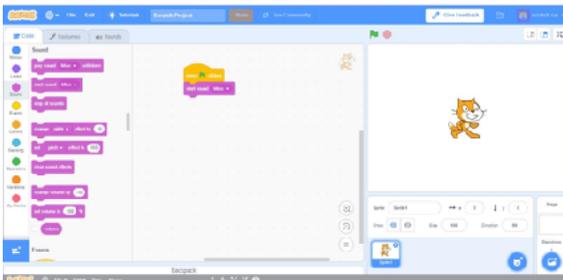
Ce sont les principales différences entre Scratch 2 et 3. Les deux versions de scratch peuvent fonctionner en ligne via des navigateurs Internet sur PC. Scratch 3 est construit en HTML5 afin qu'il puisse également fonctionner sur les tablettes Android ou iPads, mais il n'a pas de version offline contrairement à Scratch 2 qui est intégré dans Flash. Les deux programmes peuvent interagir avec des appareils externes mais uniquement sur les ordinateurs car ils ont besoin pour installer un petit programme de liaison. Scratch 2 peut contrôler nativement Lego WeDo 1 et 2 et Picoboard. Scratch 3 peut contrôler nativement Lego WeDo 2, Lego Mindstorm EV3 et Microbit, il a également des fonctionnalités supplémentaires comme la synthèse vocale dans un langage et un traducteur différents. Il est possible de créer de nouvelles extensions afin qu'à l'avenir, il y ait plus de fonctionnalités supplémentaires. Il y a d'autres fonctionnalités supplémentaires communes aux deux plates-formes comme le dessin au stylet, les instruments de musique et la détection vidéo.



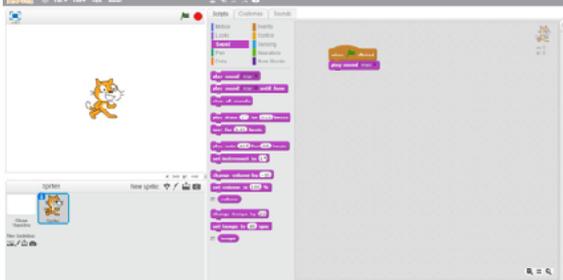
<b>Principales différences entre Scratch 2 et Scratch 3</b>		
	<b>Scratch 2</b>	<b>Scratch 3</b>
Version Hors ligne (Windows, OSX)		
Version En ligne	(seulement sur ordinateur)	(ordinateur et tablette)
<b>Périphériques externes</b>		
Lego WeDo 1	X	
Lego WeDo 2	X	X
Picoboard	X	
Lego Mindstorm EV3	X	X
Microbit		X
<b>Fonctionnalités supplémentaires</b>		
Synthèse vocale dans différentes langues	X	X
Dessin au stylet	X	X
Instruments de musique		X
Détection vidéo		X
Traduction de texte		X

Les deux versions permettent d'enregistrer et de partager des projets Scratch au sein de la communauté Scratch. Les fichiers sont compatibles avec les deux versions. Depuis janvier 2019, la seule version en ligne est Scratch 3 et la seule version hors ligne est Scratch 2.

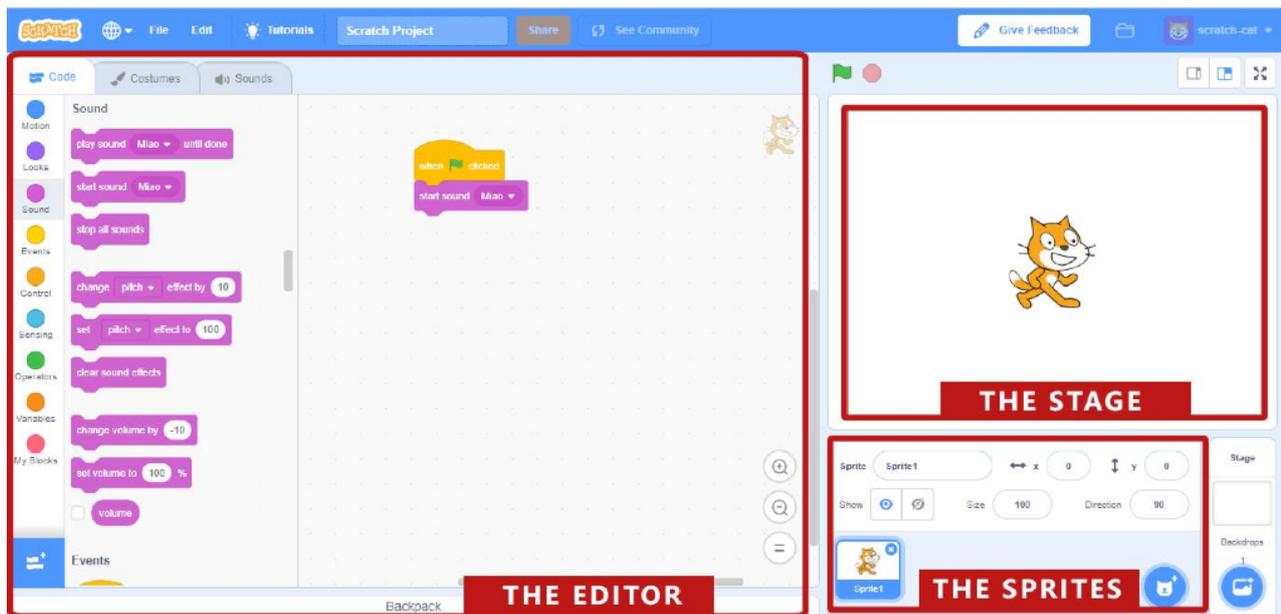
### SCRATCH 3 INTERFACE



### SCRATCH 2 INTERFACE



## L'interface



La nouvelle interface de Scratch 3 est divisée en trois zones principales :

**La scène (the stage)** est le domaine le plus important où le programme codé «prendra vie».

**La zone des sprites (the sprites)**, dans ce domaine, il y a tous les composants (sprite) qui font partie de la scène.

**L'éditeur**, il comprend trois types différents d'éditeurs.

1. **L'éditeur de code:** contient la liste des blocs disponibles et tous les blocs utilisés pour décrire le comportement du sprite respectif. Il est important de comprendre que chaque sprite a ses propres blocs, donc quand nous sélectionnerons différents sprites dans la zone du sprite, les blocs dans la zone de script changeront.
2. **L'éditeur de costumes:** permet de dessiner et de modifier l'aspect des sprites. Chaque sprite a ses propres costumes.
3. **L'éditeur de sons:** permet d'enregistrer et d'éditer les sons à utiliser dans le programme.

## Snap pour Arduino

Snap for Arduino est une modification du programme de blocs Snap !, créé par l'Université de Californie à Berkeley. Grâce à Snap (qui est continuellement développé), nous pouvons programmer facilement toutes les cartes Arduino.



Sur le site Web officiel, les fonctionnalités de Snap4Arduino sont:

- Programmation par blocs, dynamique, en direct, simultanée et parallèle
- Presque toutes les cartes Arduino prises en charge
- Utilise le firmware Firmata standard
- Brochages configurables automatiquement et abstractions matérielles de haut niveau
- Vous pouvez interagir avec plusieurs tableaux en même temps
- Versions de bureau pour les trois principaux systèmes d'exploitation
- Version en ligne qui peut se connecter aux cartes Arduino via un plugin Chrome / Chromium
- Logiciel gratuit sous licence Affero GPLv3
- Transpilation de scripts simples en croquis Arduino
- Protocole HTTP pour le contrôle à distance et la diffusion en direct du Snap ! étape
- Version en ligne de commande pour GNU / Linux embarqués

## Comment installer chaque logiciel

Pour installer Scratch 3: <https://scratch.mit.edu>

Pour installer Snap4Arduino: <http://snap4arduino.rocks>

## Focus sur Scratch 3

Scratch est spécialement conçu pour les 8 à 16 ans, mais c'est un outil très utile pour tout débutant en programmation.

Scratch est utilisé dans plus de 150 pays différents et disponible dans plus de 40 langues.

Scratch est utilisé comme langage d'introduction car la création de programmes intéressants est relativement facile et les compétences acquises peuvent être appliquées à d'autres langages de programmation tels que Python et Java.

Category	Notes	Category	Notes
Motion	Moves sprites, changes angles and changes X and Y values.	Sensing	Sprites can interact with the surroundings the user has created
Looks	Controls the visuals of the sprite; attach speech or thought bubble, change of background, enlarge or shrink, transparency, shade	Operators	Mathematical operators, random number generator, and-or statement that compares sprite positions
Sound	Plays audio files and effects. Programmable sequences are now available as an extension category named "Music".	Variables	Variable and List usage and assignment
Events	Contains event handlers placed on the top of each group of blocks	My Blocks	Custom procedures (blocks).
Control	Conditional if-else statement, "forever", "repeat", and "stop", etc.		

De plus, Scratch comprend les extensions suivantes:

- Musique
- Stylet
- Détection vidéo
- Synthèse vocale
- Traduction
- Makey Makey
- micro: bit



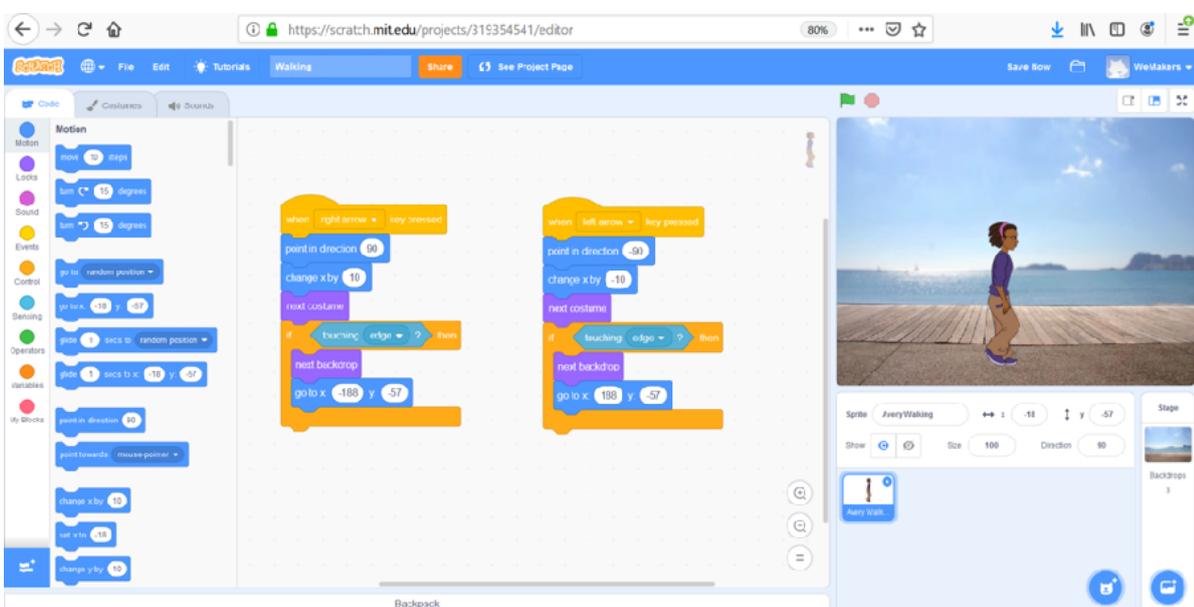
- LEGO MIDSTORMS EV3
- LEGO BOOST
- LEGO Education WeDo 2.0
- Go Direct Force & Acceleration

## Application 1: un personnage qui marche (flèches gauche / droite)

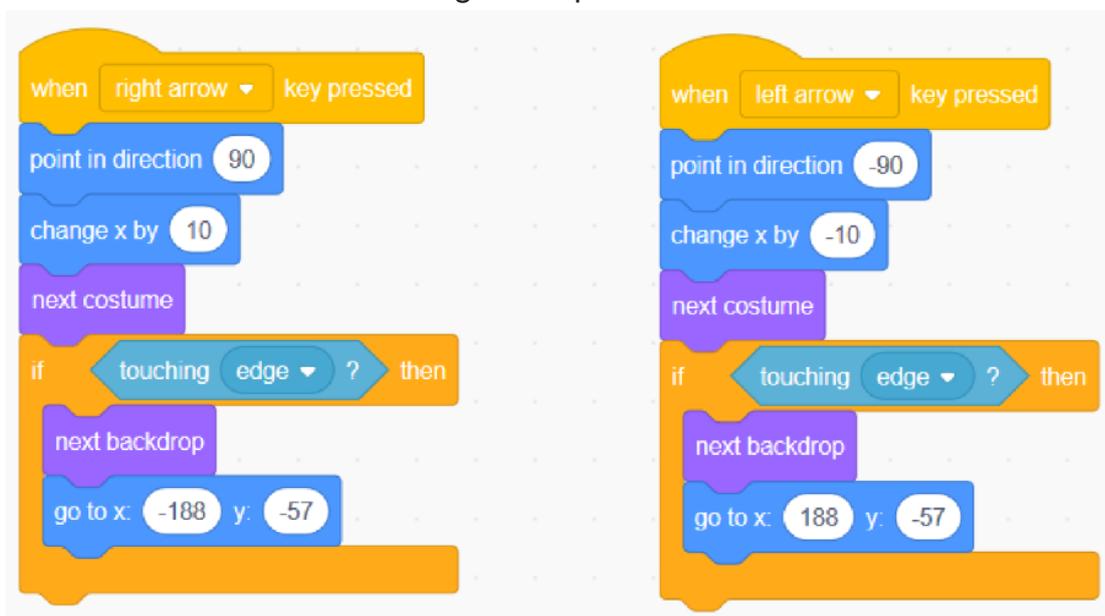
L'application suivante a un personnage (sprite) contrôlé par les flèches gauche et droite, qui marche dans trois ou plusieurs arrière-plans (arrière-plans).

Comment programmer étape par étape:

1. Choisissez un sprite (de préférence un avec des costumes de marche / vol / natation)
2. Choisissez trois décors ou plus



1. Écrivez les scripts présentés dans l'image suivante
2. Utilisez les flèches droite et gauche pour tester le résultat



## Proposition de défis

**Défi 1:** Faites sauter et avancer le personnage lorsque vous appuyez sur la flèche vers le haut.

**Défi 2:** Créez un jeu :

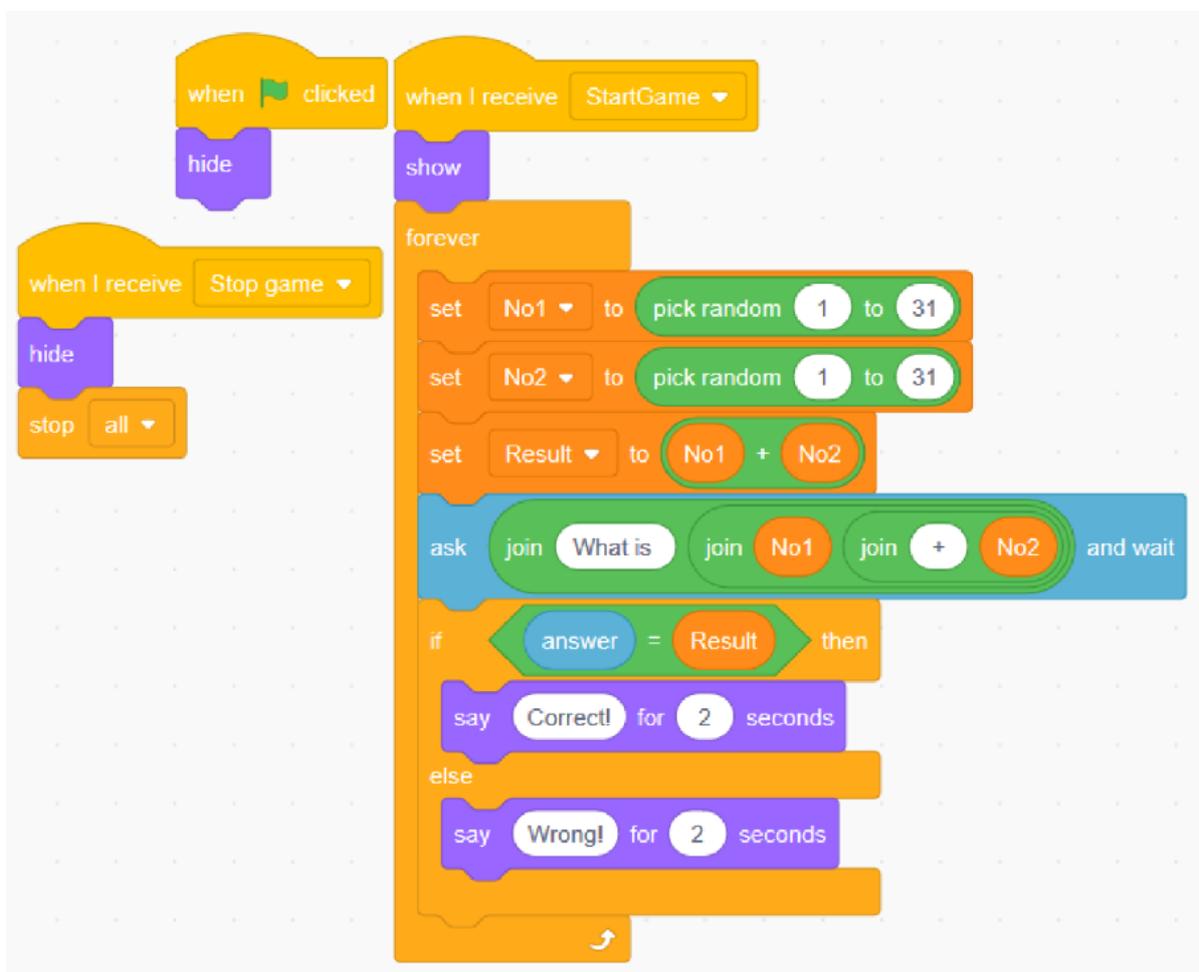
1. Un personnage contrôlable et une grenouille qui avancent au hasard (seulement changement de valeur x)
2. Le personnage ne doit pas toucher la grenouille (il peut sauter par-dessus la grenouille)
3. Il a 3 vies
4. S'il touche la grenouille, il perd une vie
5. Le jeu se termine quand il ne reste plus de vie

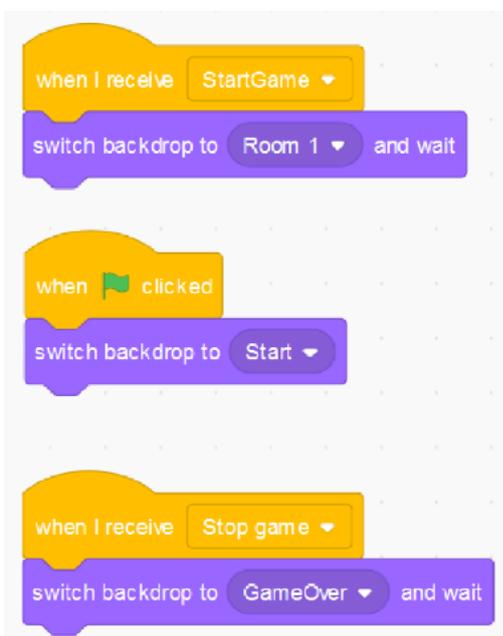
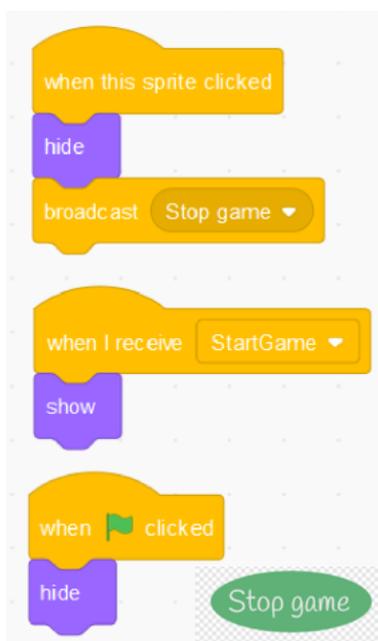
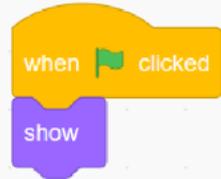
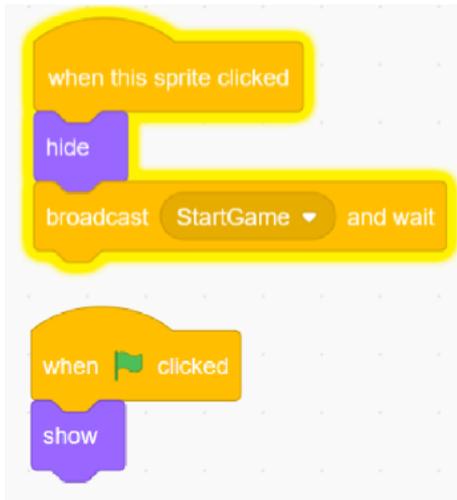
Vous pouvez ajouter le code approprié pour changer le personnage en grenouille ou autre animal pendant 2 secondes lorsqu'il touche la grenouille.

**Défi 3:** Le jeu génère des questions aléatoires et affiche Correct ou Incorrect à chaque réponse. Il a trois toiles de fond et trois sprites (2 boutons et un personnage qui pose la question et donne la réponse).



Les scripts pour chaque sprite et pour la scène sont :





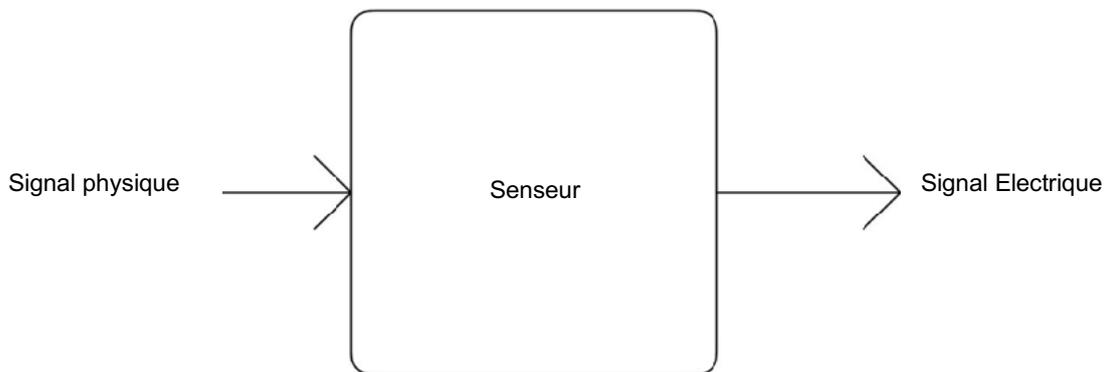
**Défi 4:** Ajoutez deux variables pour compter le nombre total de questions et le nombre total de réponses correctes. Affichez les deux valeurs à la fin.



**Défi 5:** Créez votre propre jeu de quiz avec 2-3 questions.

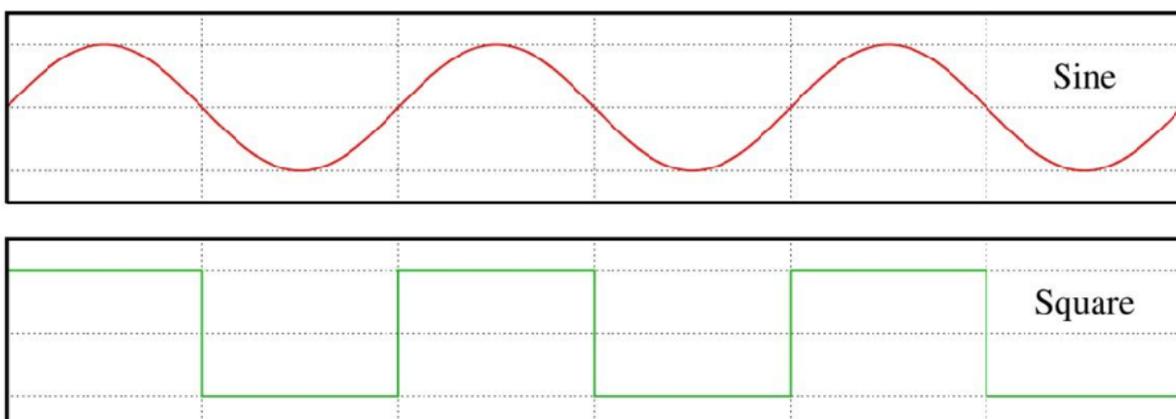
# Capteurs: un bref aperçu

Un capteur est un appareil qui est utilisé en électronique pour détecter les changements d'un paramètre physique dans l'environnement et envoyer ces informations, codifiées en un signal électrique, à d'autres appareils pour être manipulées et analysées.



Le signal électrique pourrait être analogique ou numérique. Analogique signifie que le signal varie en continuité, prenant toutes les valeurs entre un minimum et un maximum. Numérique signifie que le signal ne peut prendre qu'un nombre limité de valeurs, généralement deux: niveau bas et niveau haut.

Vous pouvez voir dans la figure ci-dessous deux exemples de signal: une onde sinusoïdale et une onde carrée. Le premier accepte toutes les valeurs de la limite inférieure à la limite supérieure, tout comme un signal analogique. Le second ne prend que deux valeurs, passant d'un état bas à un état haut, et vice versa, il s'agit donc d'un signal numérique.

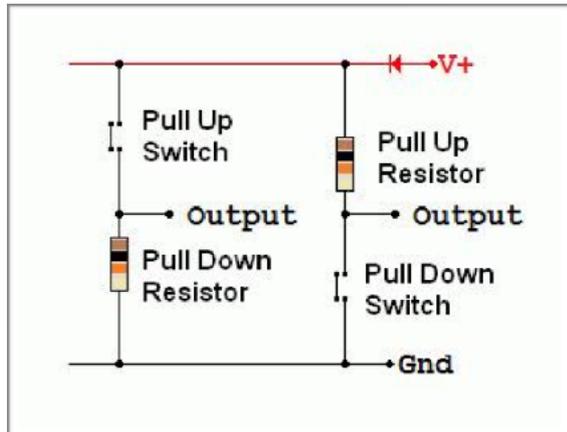




Un exemple de signal analogique est la température: il pourrait prendre toutes les valeurs du zéro absolu à l'infini. L'exemple typique de signal numérique est l'état d'un bouton: il ne peut être enfoncé ou sorti. Le capteur analogique typique est la résistance variable (par exemple le potentiomètre pour augmenter et diminuer le volume de nos amplificateurs stéréo), et le numérique est le bouton, que nous verrons dans la section suivante.

## Capteur tactile

Le capteur le plus simple que nous puissions construire est le capteur tactile. Il existe différents types de capteurs tactiles, mais le plus simple que nous pouvons utiliser est le bouton et l'interrupteur.



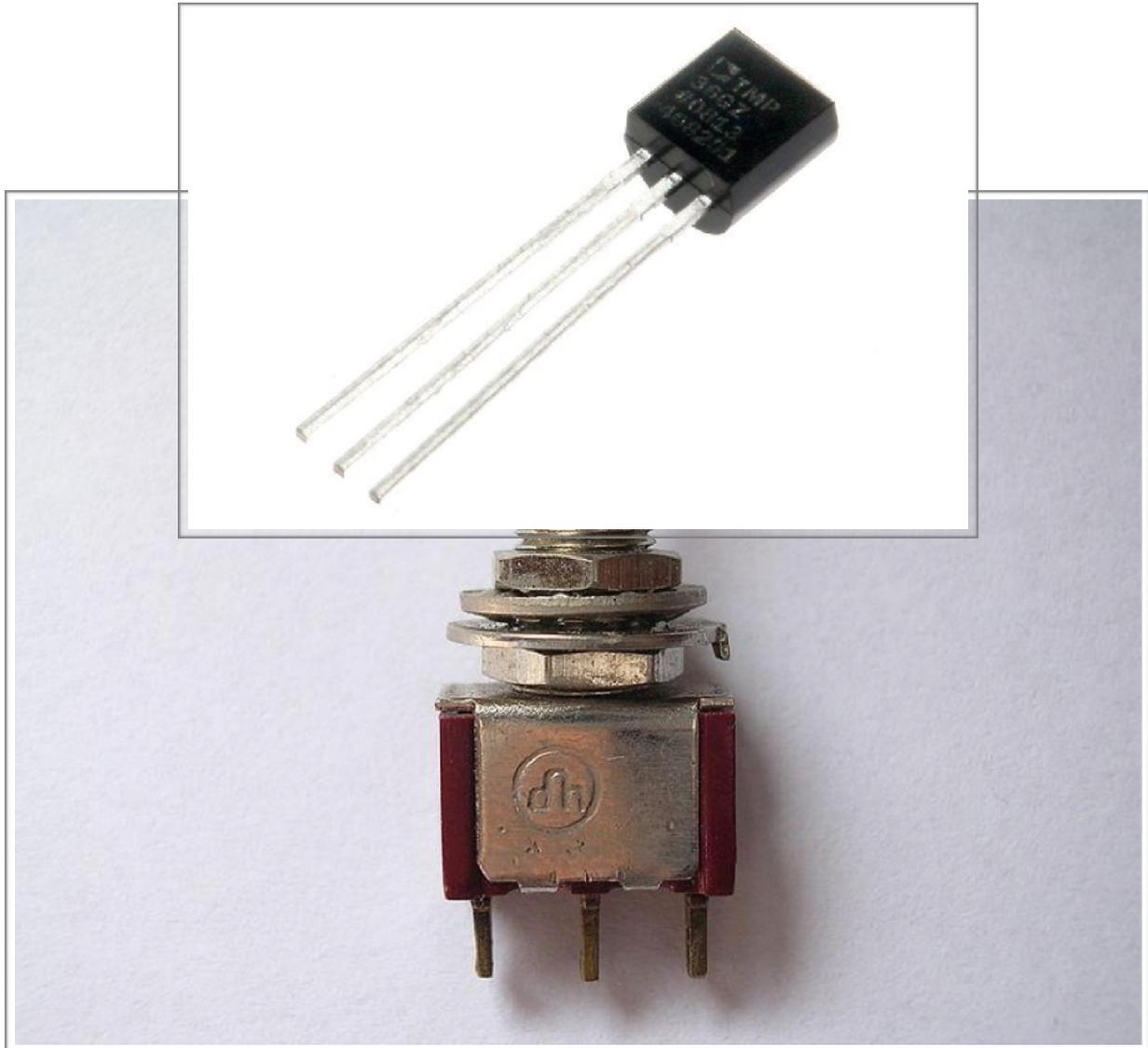
Un bouton n'est qu'un appareil électromécanique capable de fermer ou d'ouvrir un circuit électrique. Lorsque le circuit est fermé, le courant électrique peut circuler, sinon lorsqu'il est ouvert, le courant ne peut pas circuler. En plaçant un bouton dans un circuit approprié avec une résistance, nous obtenons un circuit qui ne peut donner que deux valeurs de tension, basse ou haute, correspondant à l'état du bouton. Vous pouvez voir le schéma ci-dessous pour les connexions. Vous pouvez

mettre la résistance et le commutateur de deux manières. Le premier, avec la «résistance d'abaissement» offre une valeur faible sur la sortie lorsque l'interrupteur est ouvert et une valeur élevée lorsqu'il est fermé. La version «pull up resistor» offre le comportement inverse. Par exemple, nous utilisons des boutons et des interrupteurs dans notre maison pour allumer et éteindre les lumières, ou dans l'ascenseur pour sélectionner le bon étage.



Sur ce lien, vous pouvez trouver plus d'informations sur le bouton et une carte programmable open source: <https://www.arduino.cc/en/Tutorial/Button>

Il existe d'autres capteurs tactiles, et les plus populaires sont les capteurs capacitifs. Leur technologie est utilisée dans nos appareils à écran tactile. Ils mesurent la variation de la capacité d'un condensateur due à la présence humaine. Au final, nous obtenons un appareil qui offre une sortie à deux niveaux, comme le bouton.

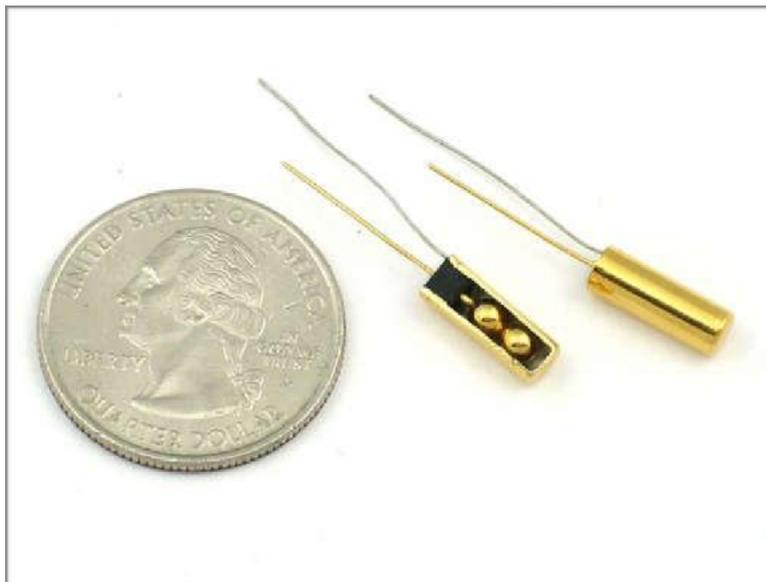


## Capteur d'inclinaison

Le capteur d'inclinaison est utilisé pour détecter si un objet est incliné dans une ou plusieurs directions. Le capteur d'inclinaison le plus simple est l'interrupteur à mercure: il est composé d'une ampoule à vide contenant deux contacts et une petite quantité de mercure, libre de se déplacer. Lorsque l'ampoule est inclinée, le mercure se déplace et lorsqu'il atteint les contacts, il ferme le circuit. Le mercure est souvent remplacé par une boule métallique. En utilisant plus de contacts, il est possible de créer des capteurs d'inclinaison multi-axes.

Un capteur d'inclinaison est utilisé, par exemple, dans certains yo-yo émettant de la lumière qui clignotent lors de la lecture ou dans un simple capteur de tangage dans certains véhicules.

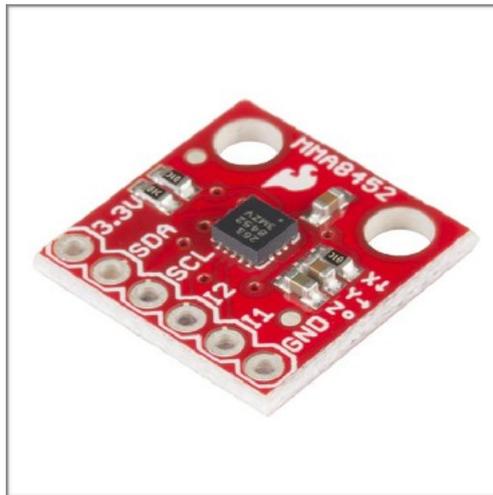
Sur ce lien, vous pouvez trouver un exemple avec un capteur d'inclinaison à axe unique et une carte programmable: <https://learn.adafruit.com/tilt-sensor/using-a-tilt-sensor>



## Accéléromètre

L'accéléromètre est un appareil utilisé pour mesurer l'accélération. Il fonctionne grâce à l'inertie physique d'une masse. Conceptuellement, un accéléromètre est composé d'une masse reliée à un ressort amorti. Lorsque l'accéléromètre est soumis à une accélération, la masse se déplace par rapport au conteneur, proportionnellement au module d'accélération. La mesure du déplacement donne l'accélération à laquelle les appareils sont soumis. L'accélération est ensuite codée en quelque sorte en un signal électrique. Les accéléromètres sont utilisés par exemple dans les systèmes de navigation pour obtenir la position relative se référant à un point zéro, dans notre smartphone comme périphériques d'entrée, pour faire pivoter l'écran, pour jouer à un jeu, comme podomètre et ainsi de suite, ou dans certains ordinateurs avec disque dur magnétique comme détecteurs de chute pour éviter les problèmes de disque dur en plaçant la tête de lecture dans une position sûre.

Sur ce lien, vous pouvez trouver un exemple d'utilisation d'un accéléromètre analogique 3 axes, qui donne trois tensions proportionnelles aux accélérations, avec une carte programmable : <https://www.arduino.cc/en/Tutorial/ADXL3xx>

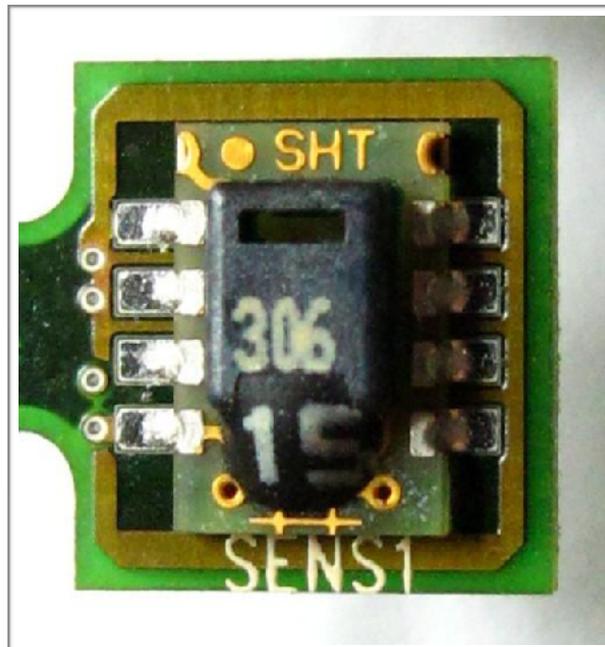


## Capteur d'humidité

Les capteurs d'humidité mesurent l'humidité dans un milieu, comme l'air ou la terre végétale. Il s'agit principalement de mesurer la variation d'une capacité et / ou d'une résistance due à la variation de la quantité d'eau dans le milieu. Les capteurs d'humidité sont utilisés par exemple en météorologie, également dans notre station portable météorologique, et dans le système d'irrigation automatique qui ne peut ouvrir l'eau que lorsque le sol est sec.

Sur ce lien, vous pouvez trouver un exemple avec une carte open source et un capteur d'humidité très courant (qui peut également mesurer la température):

[https://www.tutorialspoint.com/arduino/arduino\\_humidity\\_sensor.htm](https://www.tutorialspoint.com/arduino/arduino_humidity_sensor.htm)



## Capteur de température

Un capteur de température convertit la température en un signal électrique. Cela est rendu possible en raison de différents phénomènes physiques, selon le type de capteur. Par exemple la thermistance offre une variation de sa résistance en fonction de la variation de la température, et le thermocouple offre, pour l'effet thermoélectrique, une tension proportionnelle à la différence de la température.

Les capteurs de température sont utilisés en météorologie, thermomètres médicaux, processeurs d'ordinateurs, etc.

Ici vous pouvez voir un exemple de lecture de température à l'aide d'une carte programmable: <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temperature-sensor>

## Sorties : un bref aperçu

Une sortie est un élément qui peut agir sur le monde réel en utilisant différentes façons d'interaction. Les sorties les plus importantes utilisent le mouvement, les lumières ou le son pour interagir avec les utilisateurs et l'environnement.

### LED

Si vous souhaitez introduire des LED dans votre classe, vous pouvez utiliser de nombreuses méthodes différentes selon le type de votre école. Dans ce manuel, nous vous montrerons génériquement tous les composants électroniques dont vous aurez besoin pour que les enseignants de différentes disciplines puissent les utiliser sans aucun problème.

Avant de définir un composant, nous devons associer la création du composant à la vie de son inventeur. L'inventeur de la LED est Nicholas Holonyak Jr, un américain d'origine russe, qui a développé la première LED en 1962, il n'a pu émettre que la lumière rouge, il est aujourd'hui possible d'acheter des LED de couleurs très différentes. Par exemple, grâce Asaki, Amano et Nakamura les LED bleues ont été inventées, cette recherche était si importante qu'ils ont remporté un prix Nobel.

Une LED est une sortie apparemment très simple qui permet d'émettre de la lumière de différentes couleurs de manière fiable, elle dure longtemps et est peu coûteuse. Aujourd'hui, les LED sont utilisées pour de nombreuses applications, non seulement à des fins commerciales, mais aussi à des fins domestiques.



Le principe de fonctionnement est basé sur des électrons occupant certains espaces dans un matériau semi-conducteur et libérant de l'énergie, ou des photons. Les différentes fréquences et couleurs émises dépendent des matériaux. La LED a besoin d'un courant constant pour être alimentée, c'est la raison pour laquelle elle va toujours avec une résistance qui permet de contrôler le flux électrique qui atteint la diode.

Plus d'exercices:

Voici une liste de quelques exercices supplémentaires (non traités dans ce manuel) qui aideront vos élèves à pratiquer et à améliorer leurs connaissances, ils seront également utiles pour faciliter l'évaluation de l'enseignant:

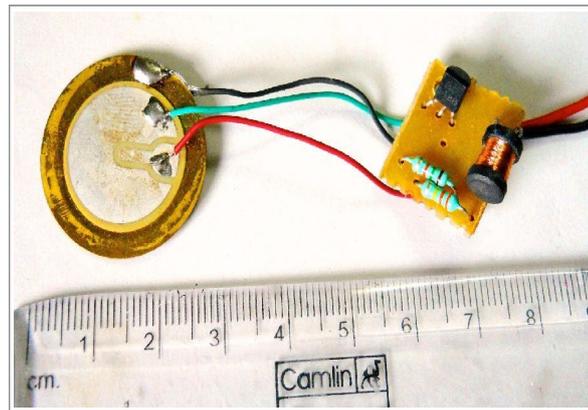
Modifiez le temps de clignotement des LED, de 1 à 3 secondes.

Changer la pin numérique (pas 0 et 1!) où la LED est connectée et changer le programme en conséquence.

Modifiez le temps de clignotement de cette façon: LED allumée pendant 1,5 seconde - LED éteinte pendant 2/3 secondes.

## Buzzer

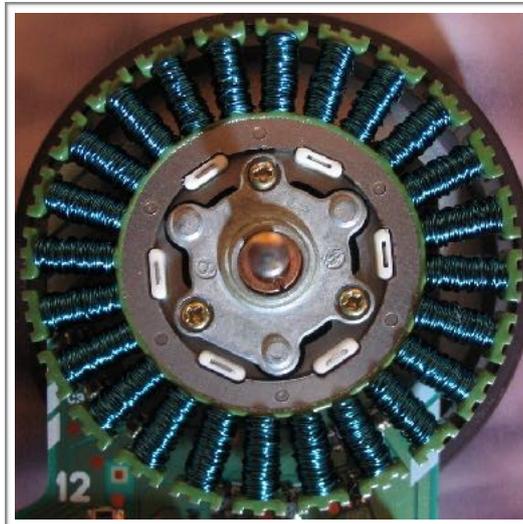
La première chose à faire avec un Arduino est de l'utiliser pour émettre des sons avec des tonalités et des longueurs différentes. Par conséquent, vous pouvez parler de la musique et des chansons et musiques préférées des élèves. En outre, il est possible de regarder une partie de "Close Encounters of the Third Kind" de S. Spielberg, celle où les humains communiquent avec des extraterrestres grâce à 5 notes de musique (l'interprétation du scientifique qui fait démarrer cette scène est celle de François Truffaut). De toute évidence, vous pouvez demander aux élèves de créer leur propre musique. De cette façon, vous pouvez impliquer les élèves à utiliser le codage de manière créative. Après avoir fait l'introduction sur la façon d'émettre des notes avec le buzzer, vous pouvez programmer le logiciel pour apprendre à jouer des chansons célèbres.



## Moteur

Les moteurs permettent de déplacer pour la première fois le robot, le premier mouvement du robot sera très simple, le robot avancera infiniment sans se soucier des obstacles et du temps de fonctionnement.

Dans le monde Arduino Le pont H est un transistor qui nous permet de contrôler deux moteurs CC simultanément. Sans le pont en H, il ne serait pas possible de faire bouger le robot. Le pont en H est en fait une batterie qui gère l'alimentation en énergie des robots. Les moteurs inclus dans le kit Arduino le plus populaire sont des moteurs à courant continu sans capteur. C'est pourquoi il n'est pas possible de gérer la rotation des moteurs mais uniquement le courant qui leur est délivré.





“Chaque fabricant de jeux vidéo sait quelque chose que les créateurs de programmes ne semblent pas comprendre. Vous ne verrez jamais un jeu vidéo annoncé comme étant facile. Les enfants qui n'aiment pas l'école vous diront que ce n'est pas parce que c'est trop dur. C'est parce que c'est - ennuyeux”

— Seymour Papert

## Défi de conception

Dans ce chapitre, nous présentons un défi de conception guidé avec différents niveaux de difficultés et différents matériels et logiciels à utiliser.

Chaque défi est pensé comme un défi à résoudre avec nos salles de classe, en utilisant certaines méthodologies pédagogiques:

- 4P: Play, Passion. Peer and Project (jeu, passion, partage et projet)
- PBL: Project Based Learning (Apprentissage par projet)
- Classe inversée

Voici une brève définition de chaque méthodologie que nous appliquerons à nos défis éducatifs:

## 4P: Project, Play, Passion and Peer (jeu, passion, partage, projet)

Mitchel Resnick présente dans le livre « Kindergarden: LifeLong learning » le concept d'apprentissage créatif.

Un étudiant créatif doit résoudre les problèmes en se concentrant sur le processus. À l'école, évidemment, nous ne pouvons pas résoudre le problème de manière drastique, mais nous pouvons apprendre un ou plusieurs processus qui pourraient permettre d'apprendre.

### **Projets :**

Un projet est un processus qui peut résoudre une série de problèmes qui composent le problème que nous voulons résoudre.

Nous n'avons pas besoin d'un projet de concert, nous pouvons également utiliser la théorie pour comprendre le monde une proposition de recherche sur l'amélioration d'un ou plusieurs problèmes.

### **Partage :**

La collaboration est fondamentale dans le processus d'apprentissage, enseigner à nos pairs, partager des informations et écouter nos collègues, nos camarades de classe nous permet d'améliorer nos connaissances et notre conscience de nos amis et de notre public. Grâce à cette partie, nous pouvons développer l'empathie.

### **Passion :**

Cela signifie Engagement. Nous pouvons travailler avec les sentiments positifs de nos étudiants lorsque nous utilisons la technologie,

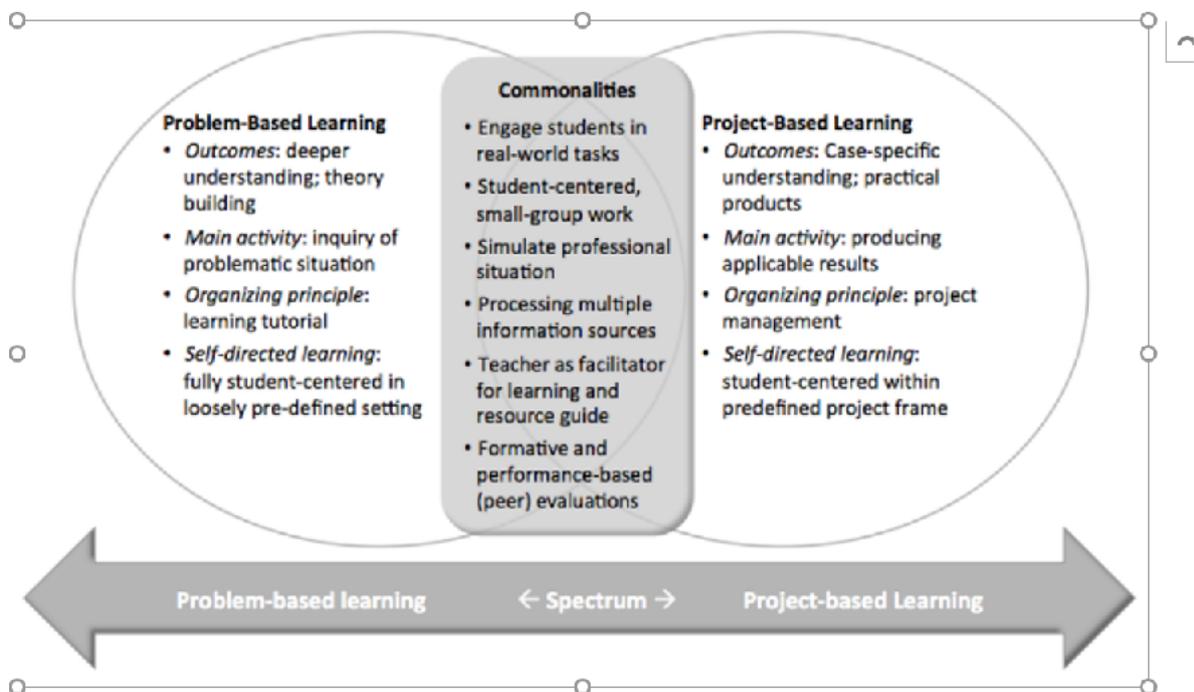
### **Jeu :**

Nous avons besoin de plaisir pendant notre processus d'apprentissage. Un environnement simple, des visages heureux permettent aux gens d'apprendre mieux et plus rapidement.

## Apprentissage par projet (PBL)

Pour encourager les apprenants créatifs, PBL est une méthodologie basée sur l'étude d'un projet qui peut impliquer toutes les disciplines. Cela signifie planifier un scénario du monde réel pour l'étudiant où les idées peuvent être liées à la résolution de problèmes réels. Grâce à la méthodologie PBL, les étudiants sont plus engagés et peuvent améliorer la collaboration, les compétences générales, favoriser la créativité et rendre l'apprentissage amusant !

Wikipedia nous apporte une image intéressante pour mieux découvrir ce qu'est le PBL:





## Classe inversée

Cette méthodologie provient de méthodes d'apprentissage mixtes et prévoit un changement de classe, où les élèves doivent étudier différentes matières et enseigner à leurs camarades de classe ce qu'ils apprennent. Dans ce processus, l'enseignant peut changer de rôle et devenir une sorte de coach, de planificateur des apprenants et de les pousser à découvrir avec le groupe de travail et à présenter de nouvelles choses

## Défi de conception 1 : baguette magique

*Kit robotique : Lego Wedo 2 ou Arduino ou Microbit (notre exemple est développé pour Lego Wedo 2)*

*Logiciel: Scratch 3.0 et Lego Digital Designer Age: de 10 à 15 ans*

### Introduction

Nous devons assembler une baguette avec des pièces de Lego WeDo 2, c'est très simple et ne nécessite pas de manuels. Pour cette raison, d'un point de vue pédagogique, la leçon peut être organisée à partir du projet de baguette à la fois sur papier et en utilisant le logiciel gratuit Lego Digital Designer.

Dans cette phase, les étudiants devront :

1. concevoir la baguette sur papier ou ordinateur (via LDD)
2. fabriquer la baguette conçue avec Lego WeDo avec le seul soin d'insérer le capteur TILT dans la poignée de la baguette.

Le but de l'exercice est de susciter l'attention et l'implication des élèves à l'aide de techniques narratives communes (histoires de magiciens). Grâce à l'utilisation de la baguette, il sera facile d'introduire l'utilisation du capteur d'inclinaison et de comprendre l'utilisation du "If" sur Scratch.

### Raconter une histoire

L'exercice se prête à travailler beaucoup sur la partie narrative, les enfants peuvent faire partie d'une histoire dans laquelle il y a des protagonistes magiciens. Ils peuvent préparer des chapeaux de magicien et les mettre pour les enfants pendant les cours. Vous pouvez travailler sur des contes de fées dédiés aux magiciens en utilisant la construction et la prochaine programmation des sorts comme une phase du récit global.

### Premier programme

Le premier programme est utilisé uniquement pour gérer la baguette et pour changer le costume d'un sprite inséré dans Scratch lorsque la baguette est déplacée.

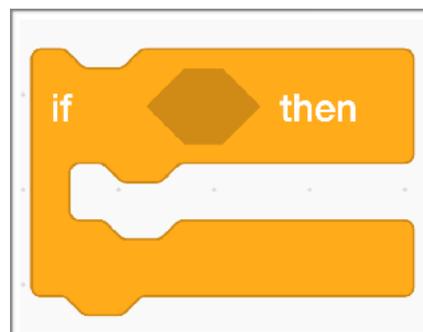
Par exemple, changer la position de la baguette de vertical à horizontal changera la forme du sprite.

Aucun pré-requis spécial n'est requis sur le langage de programmation pour exécuter ce programme. Dans cet exercice simple, les étudiants peuvent découvrir en informatique two important blocs :

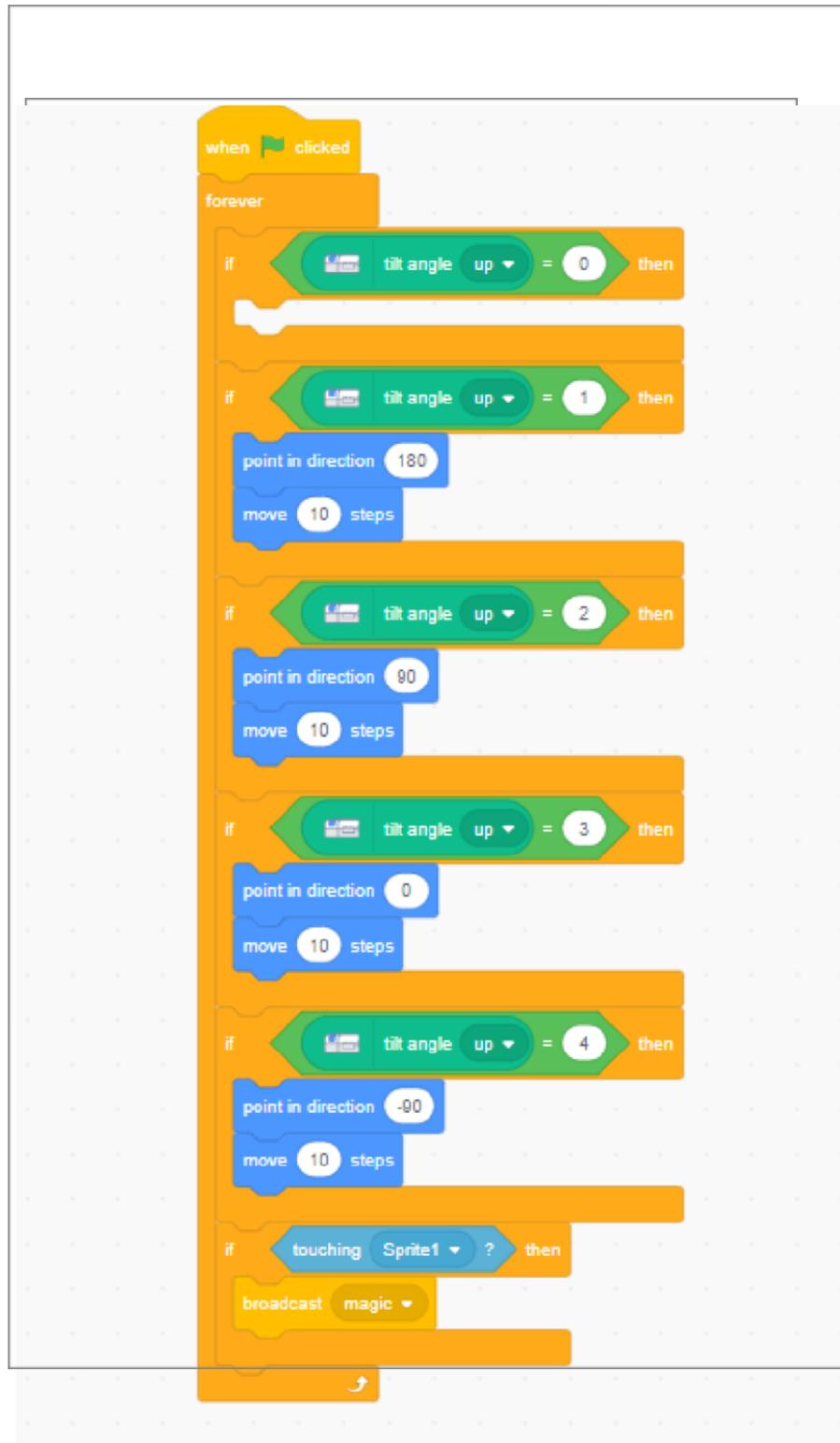


Ce bloc permet de créer une répétition de boucle infinie. Dans ce cas, cela signifie que nous contrôlons à chaque passage où se trouve la position de la baguette.

Le deuxième bloc important est IF-THEN, grâce auquel nous pouvons comparer la position du capteur et si l'angle est en hausse, nous pouvons changer l'image à l'écran.



Grâce à ce programme simple, les étudiants seront en mesure de comprendre le potentiel de la commande "IF" et de construire un organigramme simple qui comprend une ou plusieurs options basées sur l'état du capteur TILT du Lego WeDo.

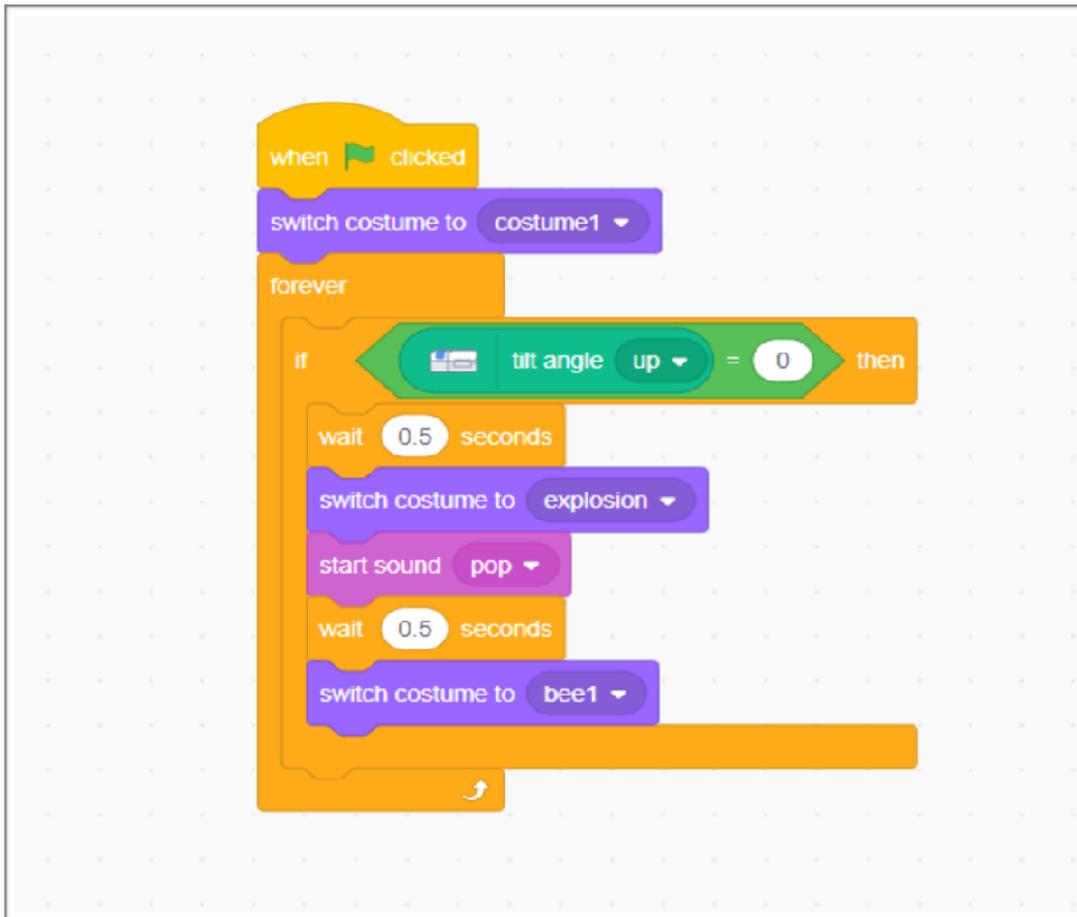


```
when clicked
  forever
    if tilt angle up = 0 then
    if tilt angle up = 1 then
      point in direction 180
      move 10 steps
    if tilt angle up = 2 then
      point in direction 90
      move 10 steps
    if tilt angle up = 3 then
      point in direction 0
      move 10 steps
    if tilt angle up = 4 then
      point in direction -90
      move 10 steps
    if touching Sprite 1 ? then
      broadcast magic
```

The image shows a Scratch script for a Lego WeDo robot. The script starts with a 'when clicked' event block. It then enters a 'forever' loop. Inside the loop, there are five 'if' blocks that check the 'tilt angle up' sensor. Each 'if' block has a corresponding 'then' block containing a 'point in direction' block and a 'move 10 steps' block. The directions are 0, 180, 90, and -90 degrees for the first four 'if' blocks, respectively. The fifth 'if' block checks for 'touching Sprite 1' and has a 'broadcast magic' block as its 'then' block.

## Deuxième programme

Le deuxième programme ne va pas étendre l'horizon pédagogique, mais consolider les concepts nouvellement acquis. Le deuxième exercice prévoit qu'au moment de l'exécution de la magie, il y a un son et un effet graphique qui annoncent la magie. Du point de vue didactique, il n'y a pas de nouveaux concepts, les enseignants peuvent donc utiliser cet exercice pour comprendre si les élèves ont compris le potentiel du logiciel.



De plus cette partie du programme permet une forte personnalisation par l'étudiant (non seulement le son mais la conception de l'effet magique). Leur personnalisation permet une plus grande implication des étudiants et un apprentissage plus grand et plus immédiat.

### Troisième programme

Sur les deux premiers exercices, vous pouvez faire de nombreuses variations sur le sujet, de difficultés égale entre elles. Pour augmenter le niveau de programmation, vous pouvez changer de cette façon la demande à l'étudiant : "créer une baguette magique qui devient le contrôleur d'une étoile sur la vidéo. Tout ce qui est touché par la baguette est transformé".

Cette demande implique la création de nouveaux algorithmes :

1. déplacement d'un sprite en fonction de la position du capteur TILT (4 états identifiés);
2. liaison d'un flux de programmation à un autre (présent dans un autre sprite) à l'aide de la commande "Diffusion".

## Défi de conception 2: poisson et capteur d'inclinaison

L'objectif est de mettre en jeu toutes les connaissances acquises jusqu'à présent dans un programme complexe, qui implique l'interaction entre de nombreux sprites. La variable utilisée (déjà présente dans Scratch) est appelée "dimension".

Cette partie est dédiée en particulier au lycée de première année et à la première année du secondaire. La structure du programme est légèrement différente des précédentes pour accompagner les enseignants dans la programmation de Scratch: il n'y aura pas d'exercices gradués, mais un seul décrit pas à pas.

### Objectif

L'objectif est de créer un jeu dans lequel un poisson prédateur (numérique ou même "réel"), commandé par le capteur Tilt, ne peut manger que les plus petits poissons. À chaque poisson mangé, le poisson prédateur grandit. L'ennemi est un poisson beaucoup plus gros, un super-prédateur, que le joueur devra éviter jusqu'à ce qu'il soit devenu assez gros pour ne pas prendre plus de risques. Ce n'est qu'alors que nous aurons atteint la fin de la partie.

### Histoire

Pour construire ce jeu vidéo, vous devez introduire certains éléments liés à l'intrigue d'une histoire. Nous devons introduire la figure du protagoniste, le but du jeu (manger d'autres poissons) et la figure de l'antagoniste, l'ennemi, qui entrave la réalisation de l'objectif.

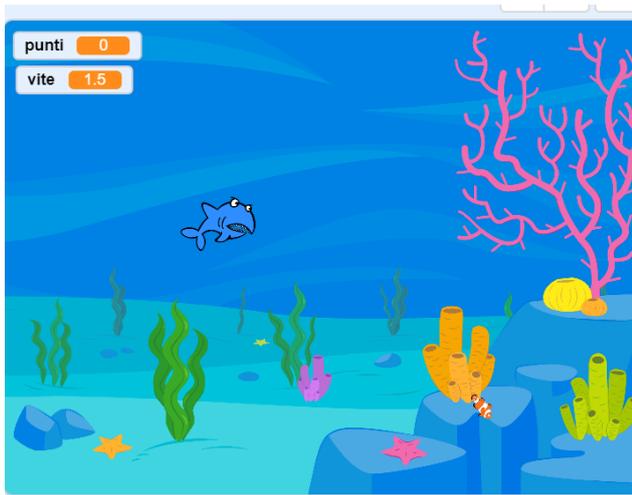
Cela peut être une première étape vers l'introduction réelle d'une histoire dans laquelle tous les acteurs sont clairement définis: protagoniste, assistant du protagoniste, antagoniste, assistant de l'antagoniste, objet magique, objectif à atteindre.

### Le jeu

Pour pouvoir réaliser le jeu vidéo, la programmation de chaque élément étape par étape et une image du jeu fini est présentée.

## Arrière-plan (scène)

Comme dans le projet 1 vous pouvez importer un arrière-plan et le modifier avec l'éditeur de dessin, ou en créer un nouveau. Dans cet exercice, il n'est pas nécessaire d'associer aucune action de fond, mais il est possible de prévoir l'alternance de différents arrière-plans ; dans ce cas, ils doivent être affichés à des moments appropriés, la planification de l'apparence.

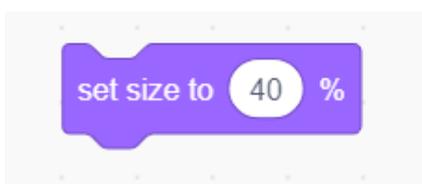


## Sprite du Protagoniste - Mouvement

Pour contrôler le sprite du poisson protagoniste, il suffit d'insérer le script dans le script principal créé dans le programme d'exercices 3 ou un autre type de script peut être :

## Sprite du Protagoniste – Initialisation de la variable "dimension"

Ce concept sous-tend tout code permettant l'initialisation des variables du programme. Dans ce cas, la variable que nous utiliserons est déjà définie dans Scratch (nous ne devons pas la créer nous-mêmes) et elle s'appelle dimension. L'initialisation se fait en utilisant la commande "set size to%" dans l'apparence du dossier (en haut à gauche). Nous définissons la dimension à 30%.



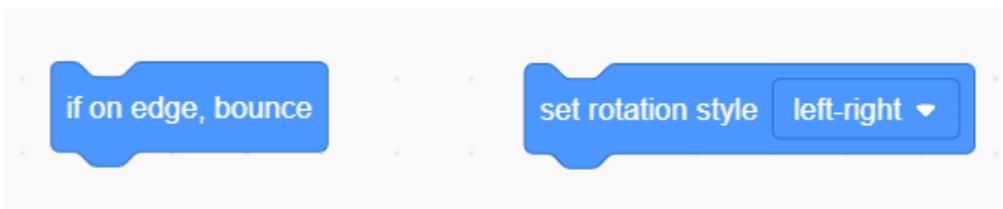
## Sprite Petits poissons - mouvement et interaction

Dans le jeu vidéo, il y aura de nombreux petits poissons, tous utilisant le même script. Il suffit de faire un "petit poisson" sprite unique, puis de le dupliquer autant de fois que nécessaire. Tous les sprites du jeu doivent être activés via le même bouton, en l'occurrence "l'Espace" (ce sont des actions qui sont activées "en même temps").

Lors de la programmation d'un jeu vidéo, vous devez penser que l'état "afficher" ou "masquer" se fait lors de l'initialisation, en particulier dans le cas où pendant le jeu vous pensez également utiliser la commande "Masquer".

Pour commencer, nous devons envisager d'afficher le sprite et de définir sa taille initiale à 20%.

Nous procédons ensuite à la programmation du mouvement et de l'interaction. Le mouvement du poisson se déroule le long d'une ligne mais vous devez penser à ajouter la commande "Rebondit lorsque vous touchez le bord", pour garantir la continuité du mouvement. Si au moment du rebond l'inversion du sprite est suffisante, sélectionnez le bouton « left-right » (comme indiqué dans l'image suivante).



## Sprite – Le Protagoniste

Pour programmer correctement, vous devez considérer que, au moment où le sprite du protagoniste est touché par un poisson, il faut évaluer si le poisson est plus gros ou plus petit. Si le protagoniste rencontre un poisson plus petit (<30 dans nos réglages ...), ce l'on découvre avec la commande "quand je reçois... taille xx", il suffira de suggérer au programme d'augmenter la taille du protagoniste d'un pourcentage de notre choix (dans l'exemple 10).

## Sprite – Gros poisson

Comme pour les petits poissons il doit être montré dès le départ. Mais contrairement à eux nous le feront apparaitre au hasard sur la scène, en utilisant une combinaison de nombres aléatoires (x, y) qui seront fournis par le script approprié comme montré ci-après.



La direction de déplacement du poisson sera également aléatoire. La taille initiale du gros poisson est fixée à 70%. Pour garantir un mouvement varié dans tout l'espace disponible, vous pouvez paramétrer le poisson pour qu'il rebondisse sur chaque bord et ait une vitesse aléatoire (le nombre de pas est généré aléatoirement de 1 à 15). Comme pour les petits poissons, si le gros touche le protagoniste il doit envoyer un message indiquant sa taille (dans ce cas 70).

### **Sprite du Protagonist - recevoir un message du grand poisson**

Lorsque le protagoniste reçoit le message du grand poisson, il faut déterminer s'il sera "mangé" ou s'il peut affronter l'adversaire, condition qui sera remplie lorsque sa taille sera d'au moins 60%. Si la condition est vérifiée, le jeu vidéo se termine par le grossissement extrême du protagoniste. Dans le cas où le protagoniste est plus petit que le grand poisson, le jeu se terminera par un message représenté par un sprite qui sera activé à la réception du message "game over".

### **Sprite Game Over - Script**

Game Over est un sprite conçu par l'utilisateur. Le mot Game Over n'est activé que lorsqu'il reçoit le message Game Over, alors qu'au début du programme il est caché.

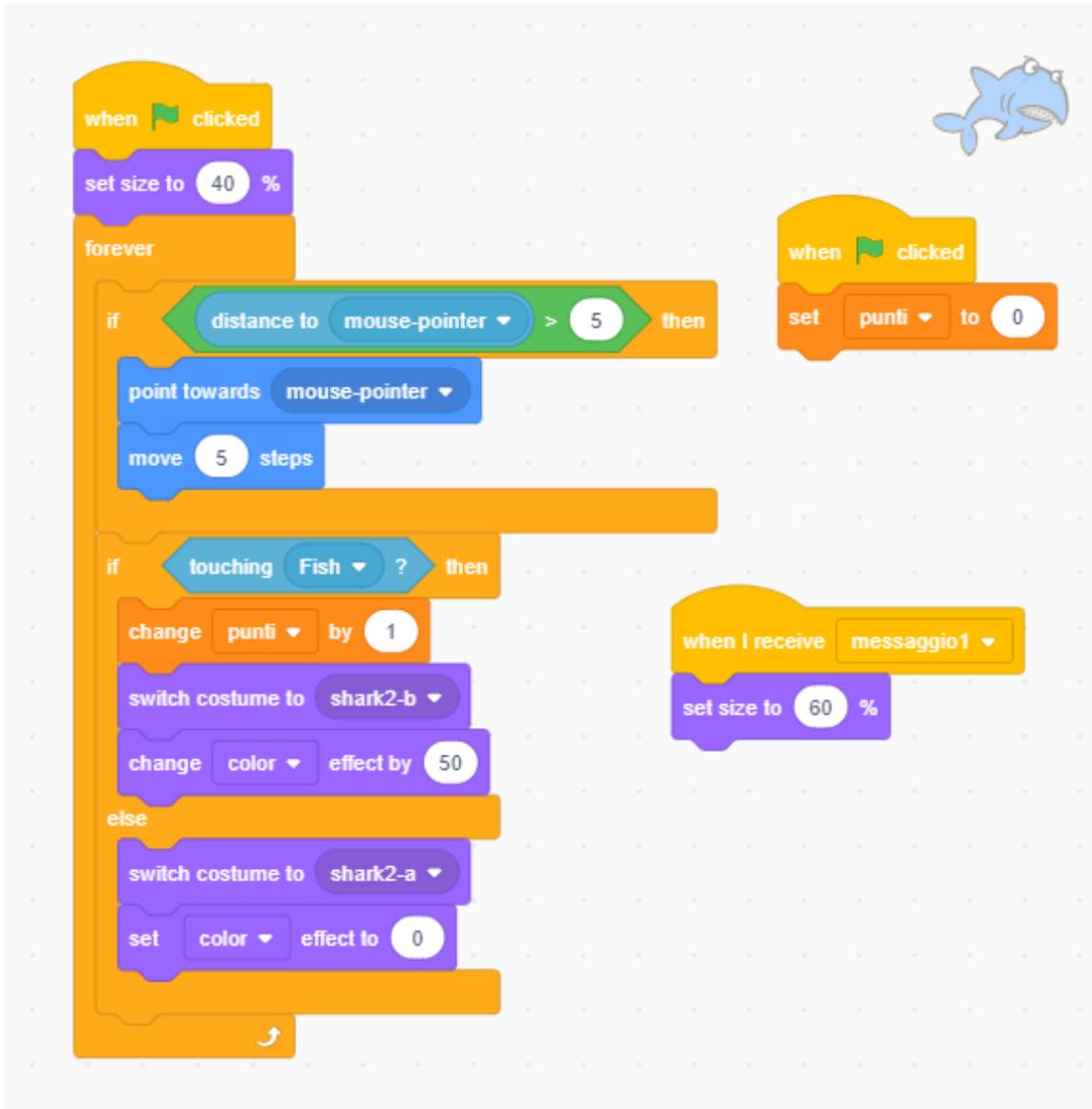
### **Sprite Gros poisson - Victoire**

Le protagoniste en cas de victoire envoie un message "gagné" à tous les sprites. Lorsque le message atteint le sprite du Grand Poisson, celui-ci doit disparaître. Si vous le souhaitez, il peut être ajouté un sprite "Vous avez gagné". Dans le programme proposé, le protagoniste devient énorme et continue de nager.

## Récapitulons

La réalisation de ce logiciel est plus complexe donc afin de ne pas alourdir la lecture nous n'avons pas inséré de notations didactiques dans le texte comme dans les exemples précédents. L'enseignant dans la conduite d'un laboratoire dédié à la réalisation de ce jeu devra clarifier certains concepts clés.

Un exemple de script simplifié pour commencer pourrait être:



The image shows a Scratch script for a shark character. The script is composed of several blocks:

- when green flag clicked** (yellow block):
  - set size to 40 %** (purple block)
  - forever loop** (orange block):
    - if distance to mouse-pointer > 5 then** (green block):
      - point towards mouse-pointer** (blue block)
      - move 5 steps** (blue block)
    - if touching Fish ? then** (green block):
      - change punti by 1** (orange block)
      - switch costume to shark2-b** (purple block)
      - change color effect by 50** (purple block)
    - else** (orange block):
      - switch costume to shark2-a** (purple block)
      - set color effect to 0** (purple block)
- when green flag clicked** (yellow block):
  - set punti to 0** (orange block)
- when I receive messaggio1** (yellow block):
  - set size to 60 %** (purple block)

A small blue shark icon is visible in the top right corner of the workspace.

```

when clicked
  set size to 20 %
  switch costume to fish-a
  hide
  wait pick random 1 to 5 seconds
  if pick random 1 to 2 = 1 then
    set x to 240
    point in direction -90
  else
    set x to -240
    point in direction 90
  set y to pick random -180 to 180
  show
  forever
    turn pick random -10 to 10 degrees
    move 3 steps
    if on edge, bounce
    wait 0.01 seconds
    if touching Shark 2 ? then
      wait 0.05 seconds
    if touching Shark 2 ? then
      wait 0.05 seconds
      broadcast messaggio1
      start sound ocean wave
      hide
      wait pick random 1 to 5 seconds
      if pick random 1 to 2 = 1 then
        set x to 240
        point in direction -90
      else
        set x to -240
        point in direction 90
      set x to pick random -180 to 180
      show
  
```



```

when clicked
  set vite to 2
  forever
    if touching Shark 2 ? then
      set vite to vite - 0.5
      go to x: -183 y: -121
      wait 0.5 seconds
    if vite = 0 then
      switch costume to fish-c
  
```

## **La variable "dimension"**

Scratch permet de créer de nouvelles variables mais certaines sont préexistantes et définies. "Dimension" est l'exemple classique d'une variable locale, qui ne peut s'appliquer qu'au sprite pour lequel elle est utilisée, contrairement à d'autres variables qui peuvent être utilisées par plusieurs sprites.

L'utilisation de cette variable ne permet pas par exemple de la rappeler dans les scripts d'autres sprites.

Grâce à la "dimension", le concept de variable peut être expliqué. Dans ce jeu, il est clair que le changement de la dimension variable diminue ou non le succès du

Jeu. Très souvent, lorsque les variables sont introduites à l'école, ce sont des exemples créés ad hoc pour le cours, peu fonctionnels ou trop complexes. L'utilisation dans ce jeu vidéo offre une introduction souple des variables.

## **Nombres aléatoires**

Pour générer des poissons qui se déplacent à des vitesses aléatoires, vous devez utiliser le générateur de nombres aléatoires Scratch, cela nous permet de faire comprendre aux étudiants comment ces «générateurs» sont utilisés dans le monde de l'informatique.

## **Envoyer et recevoir**

Dans ce programme, la communication entre les sprites est essentielle. Cela nous permet d'introduire des flux parallèles et de programmation pour créer des organigrammes préparatoires complexes.

## Défi de conception 3: L'Oracle

*Logiciel: Scratch 3*

Ce défi fait le lien avec le monde des assistants vocaux, qui représentent un marché important. Toutes les grandes entreprises technologiques (Apple, Google, Amazon, Microsoft...) proposent un assistant vocal.

Pour commencer la leçon, nous pouvons inspirer nos étudiants en commençant par un film de science-fiction, 2001 l'Odyssée de l'Espace et HAL2001 l'assistant vocal qui interagit avec les astronautes et contrôle le vaisseau spatial. Nous pouvons également citer le monde de Star Trek, où tout l'équipage peut parler avec le «navire» pour demander de l'aide, des suggestions et un service.

Enfin nous pouvons revenir dans le monde réel en présentant tous (ou presque) les produits réels du marché, dont voici quelques exemples :

### Alexa (Amazon)



### Google Home (Google)



## Siri



Tous ces assistants vocaux peuvent répondre à vos questions en vérifiant les informations sur Internet et en utilisant les liens d'applications les plus importants vers les plate-forme de musique en ligne, les services cloud, le marché en ligne.

D'un point de vue éthique, nous pouvons citer avec les étudiants L'épisode « Rachel, Jack and Ashley Too » de la série « Black mirror ». L'utilisation de la voix d'un chanteur célèbre peut amener dans certains cas une mauvaise utilisation de cet assistant vocal qui se substitue aux vrais amis de l'adolescent. Le film « Her » est une autre illustration de cette possible dépendance.

D'un point de vue éthique, il est important de présenter des exemples et de les utiliser avec les étudiants pour introduire le concept d'éthique en technologie.

La présentation de l'assistant vocal peut également être utilisée pour introduire des concepts d'IA.

## Assistant vocal et Internet des objets

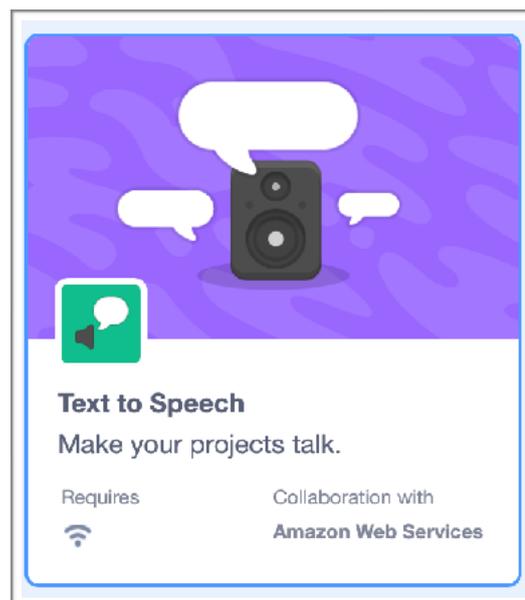
Nous avons présenté l'utilisation de l'assistant vocal seul, mais dans un avenir proche, cet assistant pourrait devenir l'interface principale de tout ce qui est connecté à Internet dans notre vie quotidienne.

Après cette introduction, nous pouvons découvrir comment programmer un assistant vocal avec Scratch 3:

Il n'est pas possible dans Scratch de base d'utiliser la reconnaissance vocale, mais nous pouvons créer un programme où l'interaction se fera avec la voix de l'ordinateur et le texte de l'utilisateur humain.

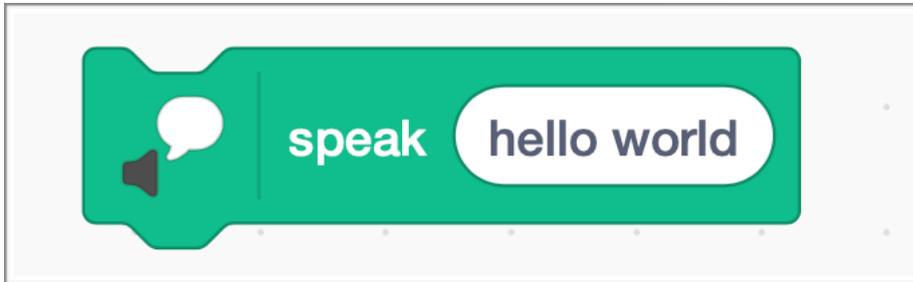
Pour programmer un assistant vocal, une sorte d'oracle, on peut utiliser la commande "Text to speech"

Dans Scratch cette fonctionnalité est une extension. Nous devons cliquer sur «plus d'extensions» et nous pouvons sélectionner cette image:



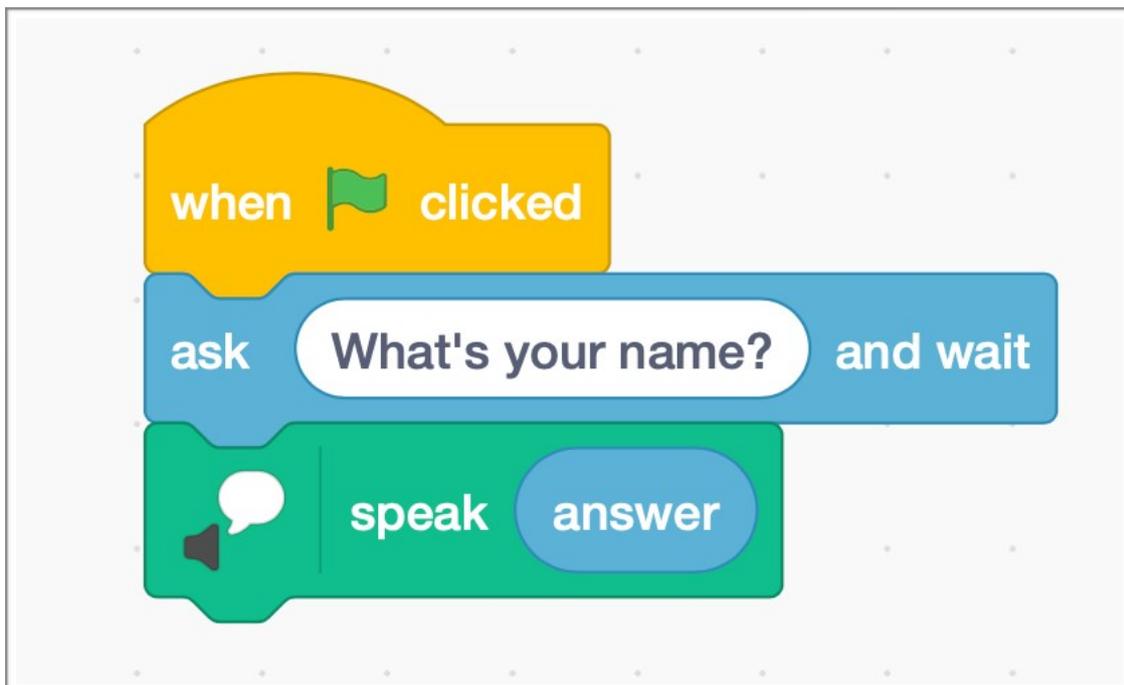
## Qu'est-ce que la synthèse vocale?

Ce bloc permet à l'ordinateur de lire ce que nous écrivons, donc l'interaction sera intéressante.



Nous pouvons utiliser les blocs suivants

Grâce à ce programme, nous pouvons demander à l'utilisateur d'écrire son nom en utilisant le clavier. En utilisant Text to speech, le programme est capable de lire le nom. Nous pouvons améliorer le programme en demandant certaines choses et en recevant des réponses différentes.



## Focus sur micro: bit



## Scratch et micro:bit

Micro: bit est un micro-ordinateur de poche conçu pour aider les enfants à apprendre à coder et à créer avec la technologie. Il possède de nombreuses fonctionnalités, notamment un écran LED, des boutons et un capteur de mouvement. Il peut être connecté et programmé avec Scratch, MakeCode et Python.

Le micro: bit BBC contient un accéléromètre qui peut détecter s'il est secoué ou de quelle manière le micro: bit est maintenu. En plus de détecter s'il est à l'envers, les accéléromètres peuvent détecter certaines des forces qui agissent sur lui. Le micro: bit a un capteur de température intégré qui peut détecter la température actuelle de l'appareil, en degrés Celsius. La liste complète des fonctionnalités est présentée dans le tableau suivant:

Feature	Description
2 buttons	Programmable action push buttons
25 LED lights	Can be individually programmed to show shapes, text or numbers
USB connector	Connect to a computer for power or to load programs onto the <a href="#">micro:bit</a>
Accelerometer	Senses if the <a href="#">micro:bit</a> is being moved, tilted, shaken, or in free-fall, and at what acceleration
Compass	Detects which direction the <a href="#">micro:bit</a> is facing
Processor	Where the program is executed
Radio	Communicate with other <a href="#">micro:bits</a> for multiplayer games
Bluetooth antenna	Wirelessly sends and receives signals to Bluetooth enabled PCs, Smartphones, or Tablets
Reset button	Restarts the <a href="#">micro:bit</a>
Battery socket	Power the <a href="#">micro:bit</a> using batteries
Temperature sensor	Detects the current temperature of the <a href="#">micro:bit</a> in degrees Celsius
Light sensor	The LEDs on the <a href="#">micro:bit</a> can also act as a light sensor to detect ambient light
Edge Connector	25 external connectors, called Pins, on the edge of the <a href="#">micro:bit</a> allow you to connect to other electronics hardware, including LEDs, motors, and other sensors. These can behave as inputs or outputs.

Pour le moment, Scratch 3.0 n'a pas de blocs de code pour utiliser la boussole, le thermomètre ou le capteur de lumière. De plus, seulement 3 des 25 broches sont utilisées.

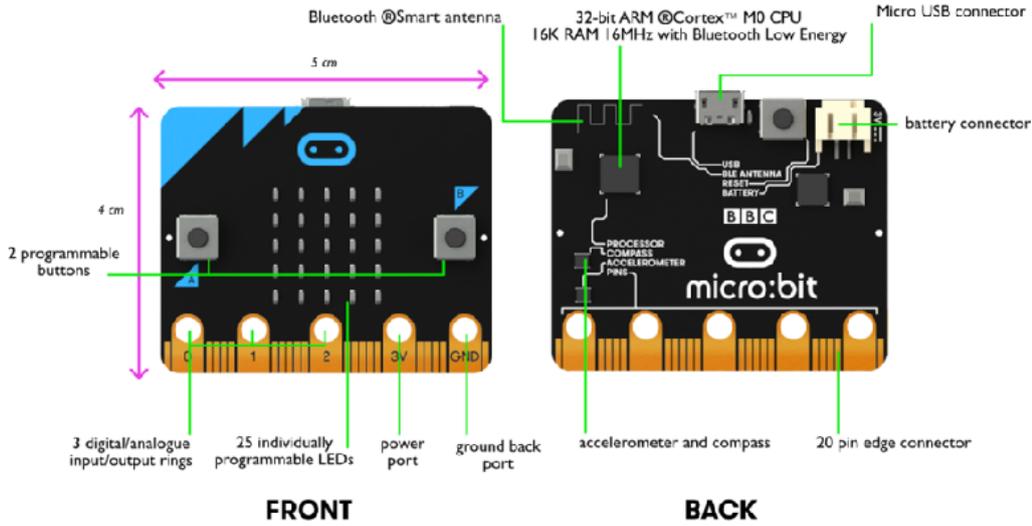


Image source: <https://www.pakronics.com.au/pages/microbit-in-australia>

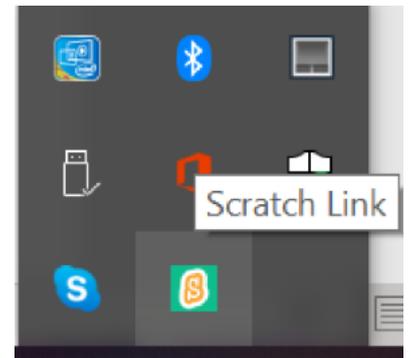
## Comment l'utiliser

Micro: bit peut être programmé sur un ordinateur de bureau (Mac, PC, Chromebooks, Linux, y compris Raspberry Pi) et mobile. Pour utiliser micro: bit avec Scratch, les étapes suivantes doivent être suivies (pour Windows).

Étape 1: installez Scratch Link depuis la boutique Microsoft ou utilisez les liens de la page <https://scratch.mit.edu/microbit>

Étape 2: Démarrez Scratch Link et assurez-vous qu'il fonctionne. Il devrait apparaître dans votre barre d'outils.

Étape 3: Connectez le micro: bit à votre ordinateur avec un câble USB. Le micro: bit apparaîtra sur l'ordinateur comme un lecteur appelé «MICROBIT».



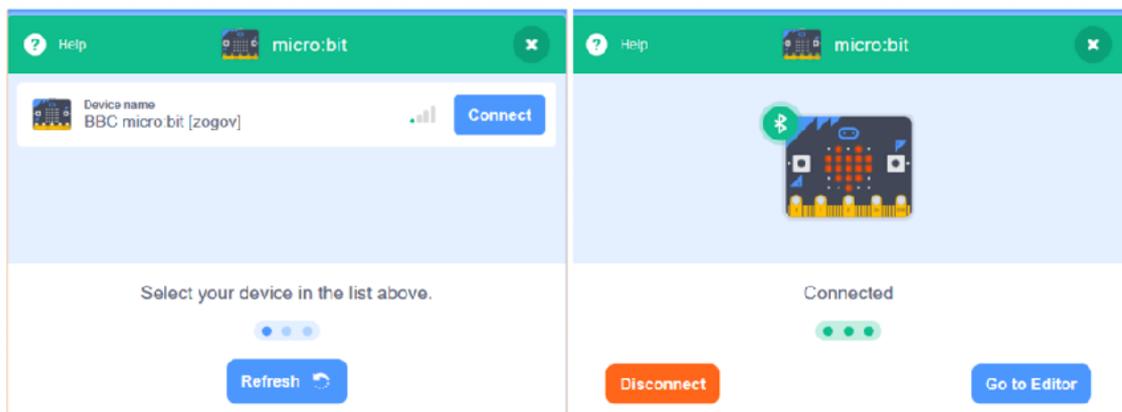
Étape 4: Téléchargez à partir de <https://scratch.mit.edu/microbit> le fichier HEX Scratch micro: bit

Étape 5: Décompressez l'archive et faites glisser et déposez le fichier HEX sur votre micro: bit Étape 6: Alimentez votre micro: bit avec USB ou une batterie.

Étape 7: utilisez Scratch Editor

Étape 8: Ajoutez l'extension micro: bit

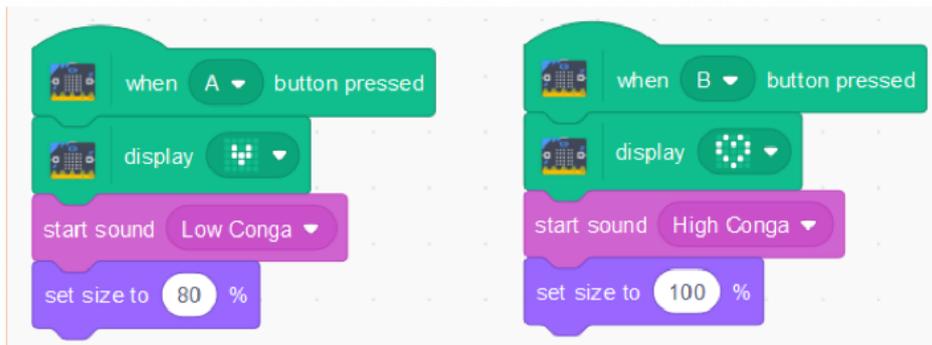
Étape 9: Appuyez sur Connecter et accédez à l'éditeur



# Défi de conception

## Défi 1: Un battement de cœur

L'application (téléchargée sur <https://scratch.mit.edu/microbit>) est contrôlée par les deux boutons de micro: bit. Selon le bouton enfoncé, il affiche sur le micro:bit les deux cœurs (un plus petit et un plus grand) et, en même temps, dans Scratch, un cœur change de taille et deux sons différents sont joués.



## Défi 2: Balle

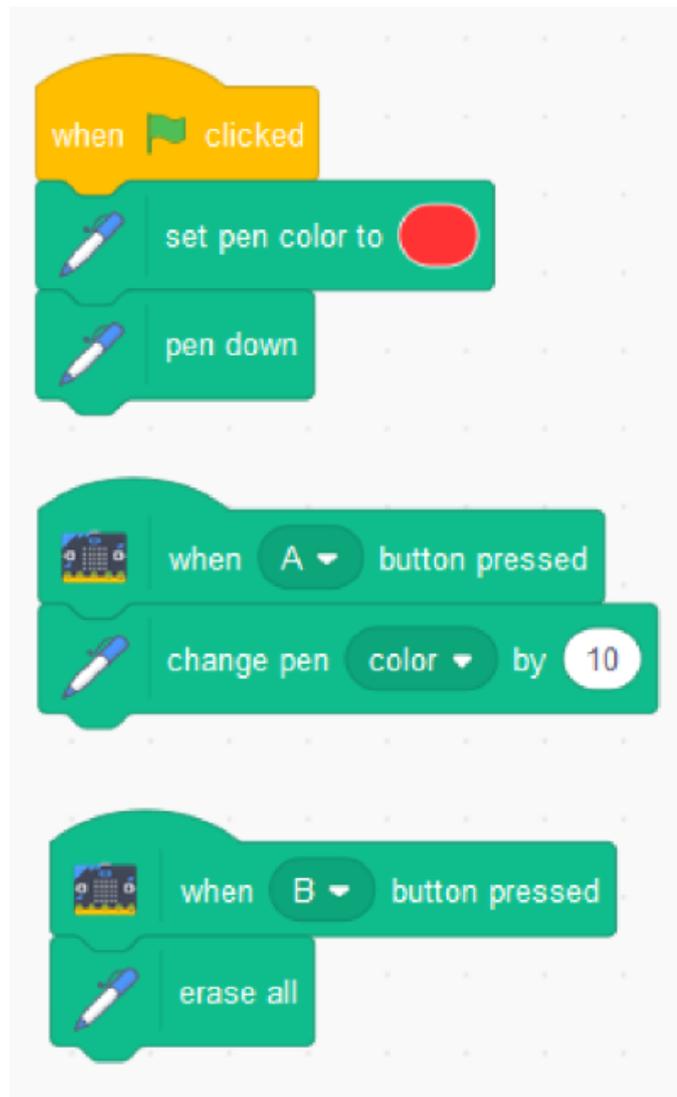
En changeant l'inclinaison du micro: bit, une balle peut être déplacée sur l'écran.

```
when clicked
  forever
    if tilted front ? then
      change y by 10
      if on edge, bounce
    if tilted back ? then
      change y by -10
      if on edge, bounce
    if tilted left ? then
      change x by -10
      if on edge, bounce
    if tilted right ? then
      change x by 10
      if on edge, bounce
```

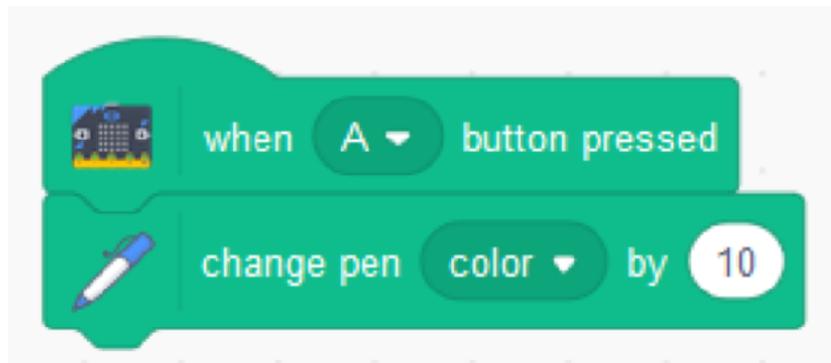


### Défi 3: Un Stylet stylé

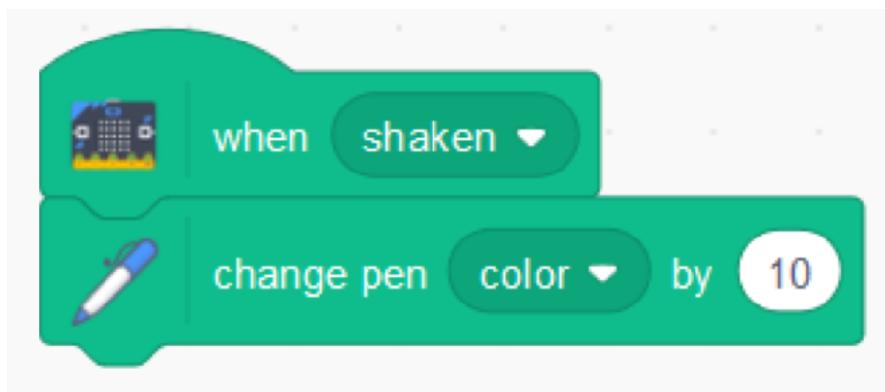
Ajoutez l'extension du stylet à Scratch. À l'application précédente, ajoutez les scripts suivants. Testez le résultat



Si vous remplacez



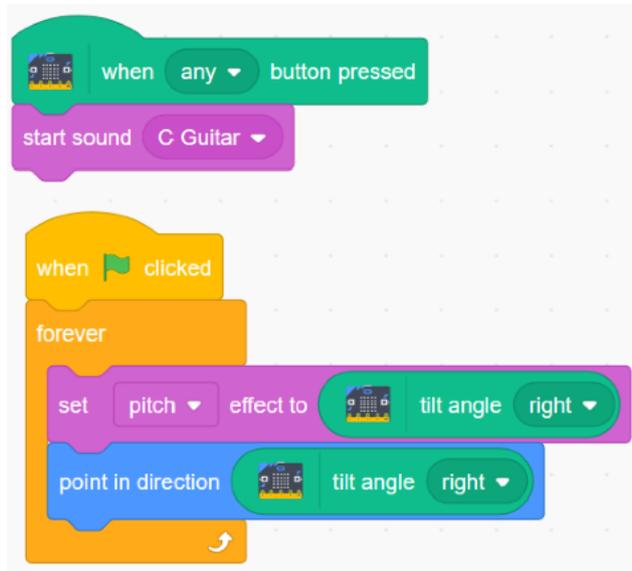
par



alors la couleur changera automatiquement lorsque le Micro:bit est secoué.

## Défi 4: Jouer de la guitare

L'application émet un son lorsque vous appuyez sur n'importe quel bouton du micro: bit avec un effet de hauteur en fonction de l'inclinaison du micro: bit.



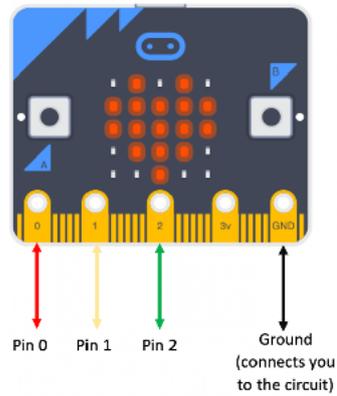
## Défi 5: Jouez d'un piano maison

Afin de créer la prochaine application (inspirée d'un exemple de <https://microbit.org/scratch/>), vous aurez besoin de la puce micro:bit, de 4 câbles avec pinces crocodiles, de papier d'aluminium, ciseaux, colle et papier.

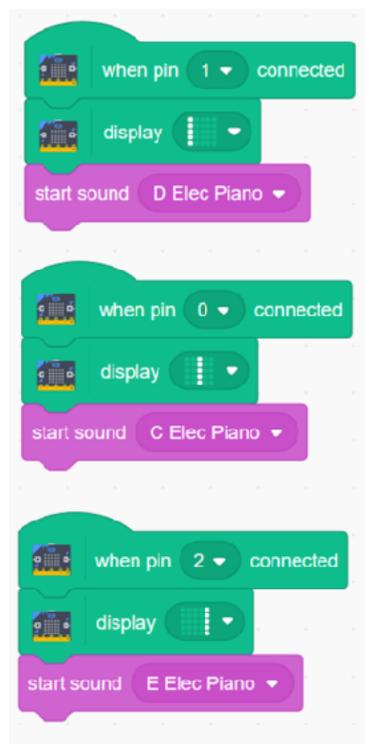
Construisez un piano comme celui de l'image suivante. Connectez chaque broche de 0 à 2 et la masse à une bande de papier d'aluminium.



Le schéma de ce piano est dans l'image suivante :



Ecrivez les scripts :



Testez l'application en touchant le piano (bandes de papier d'aluminium) (un doigt doit toucher la bande de papier d'aluminium reliée à la terre afin de fermer le circuit).

## Plus de défis

Créez un jeu de "pierre, papier, ciseaux". L'application doit afficher au hasard une pierre, un papier ou des ciseaux lorsque vous secouez le micro: bit.

Créez votre propre appareil portable avec une interface dans Scratch. Concevez un étui / support pour votre appareil et imprimez-le avec l'imprimante 3D.

## Ressources

<https://microbit.org/scratch/>

<https://ilk.github.io/microbit-extension/iste18/>

<https://make.techwillsaveus.com/>

<http://libraryadventuring.blogspot.com/2018/10/coding-and-making-with-bbc-microbit.html>

<http://blog.sparkfuneducation.com/five-wearable-projects-with-microbit>

<https://scratch.mit.edu/discuss/youtube/44Xo76Bbqil/>

## Références

<https://microbit.org/scratch/>

[https://diyodemag.com/education/kids\\_coding\\_scratch\\_30\\_meets\\_micro\\_bit](https://diyodemag.com/education/kids_coding_scratch_30_meets_micro_bit)

# MakeCode pour micro:bit

Microsoft MakeCode est un environnement Web pour apprendre à coder avec des appareils informatiques physiques tels que le micro:bit. MakeCode est gratuit et fonctionne sur toutes les plateformes et navigateurs.

MakeCode est disponible en tant qu'éditeur en ligne pour:

- micro:bit - <https://makecode.microbit.org/>
- Circuit Playground Express - <https://makecode.adafruit.com/>
- Minecraft - <https://minecraft.makecode.com/>
- LEGO® MINDSTORMS® Education EV3 - <https://makecode.mindstorms.com/>
- Arcade - <https://arcade.makecode.com/>
- Chibi Chip - <https://makecode.chibitronics.com/>

Et comme application pour :

- micro:bit
- Adafruit
- Cue de Wonder Workshop

L'application MakeCode pour micro:bit possède quelques fonctionnalités supplémentaires par rapport à l'éditeur en ligne. Avec l'application de bureau, le micro:bit peut être programmé directement via USB, sans avoir à glisser-déposer le fichier sur le lecteur micro:bit et lire directement les données série de micro:bit.

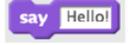
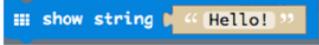
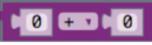
## Scratch ou MakeCode pour micro:bit

Sur la page de support micro:bit

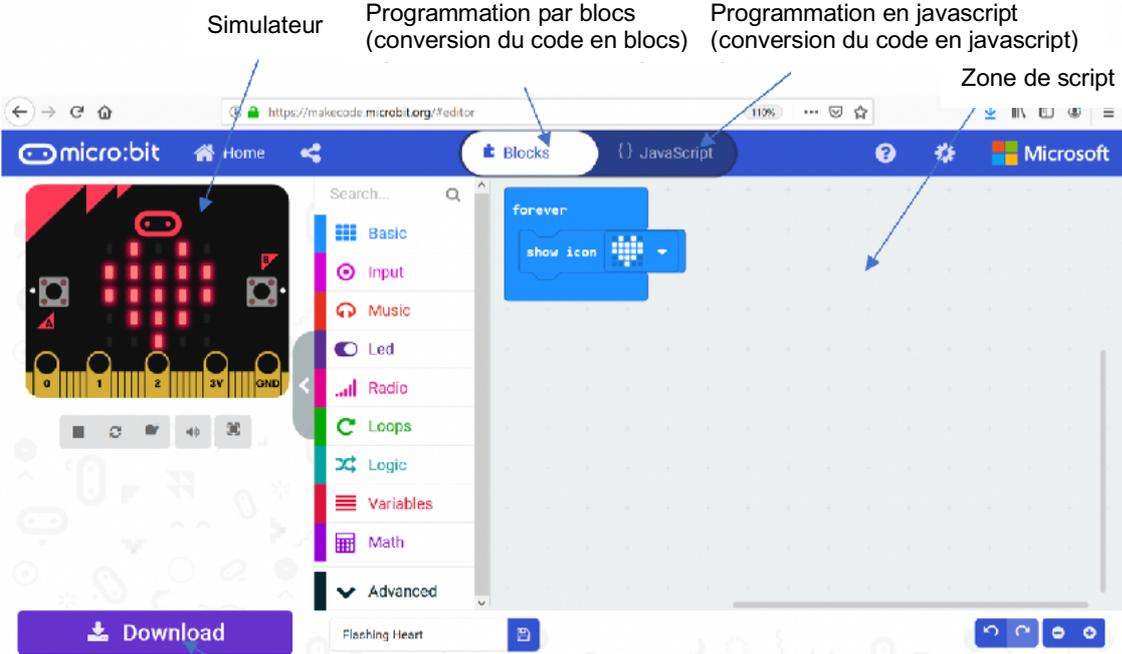
<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode>

il est publié un tableau de comparaison, répertorié ci-dessous, entre Scratch et Makecode pour micro:bit, afin de mettre en évidence les subtiles différences entre plusieurs blocs importants. Le tableau pourrait être utile à l'utilisateur de l'un des deux éditeurs en ligne, passionné de micro:bit, pour commencer à utiliser l'autre.

Un gros avantage de Scratch, par rapport à MakeCode en ligne, est qu'il permet la programmation OTA (Over-the-Air) par communication Bluetooth. Le gros inconvénient est que Scratch fournit un ensemble de blocs de base basiques pour micro:bit, seulement 10. Pour le moment, l'ensemble plus riche de blocs micro:bit dans MakeCode le rend préférable.

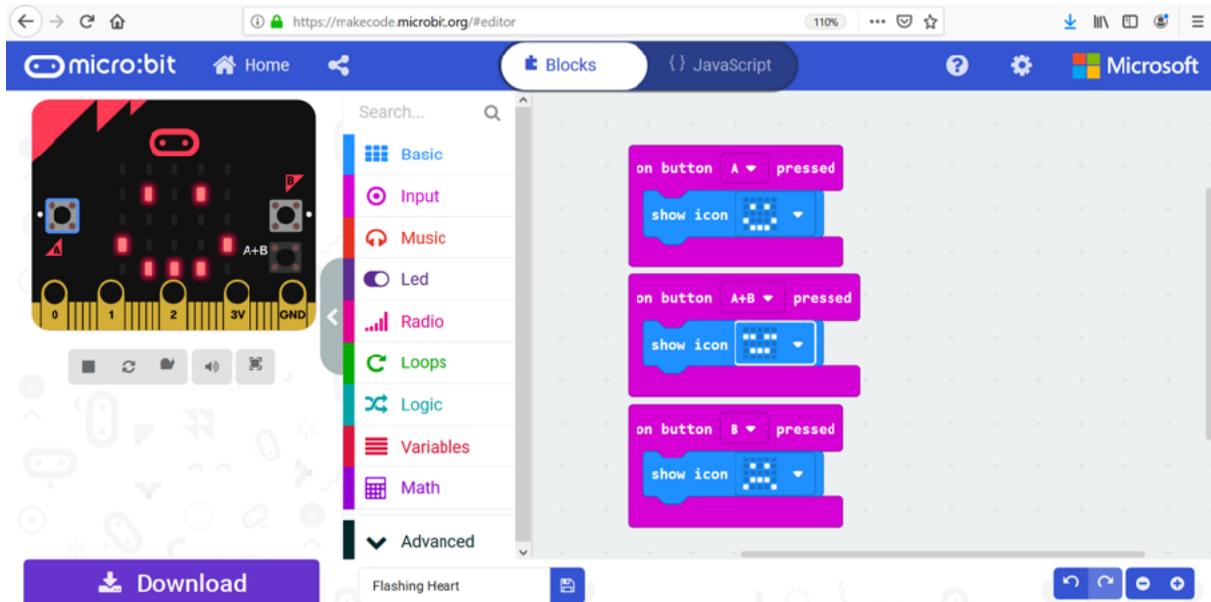
Scratch	Makecode
<p><b>Events</b> wait for a user action, like clicking the <b>green flag</b> in scratch or pressing a button on the keyboard</p> 	<p>Input waits for a user <b>input</b> like pressing the A button or shaking the <b>micro:bit</b></p> 
<p><b>Control</b> is about the flow of your program tasks. In scratch you can add a <b>forever</b>, <b>repeat</b> or <b>if</b> block to an event to trigger it.</p> 	<p><b>Makecode</b> breaks controls into programming concepts. <b>Forever</b> is it's own loop (it's triggered as soon as the <b>micro:bit</b> is powered on).</p>  <p><b>Repeats</b> are found in the <b>Loops</b> menu of <a href="#">Javascript</a> Blocks</p>  <p>And <b>if</b> blocks are found in the <b>Logic</b> menu</p> 
<p><b>Operators</b> let you do arithmetic and make comparisons</p> 	<p><b>Arithmetic</b> and random number pickers can be found in <b>Math</b>, whereas a comparison between something and something else is found in <b>Logic</b></p> 
<p><b>Data</b> lets you define variables that might change within your program. Here we have made a variable called <b>item</b></p> 	<p>We can define these in the <b>Variables</b> menu of <a href="#">makecode</a></p> 
<p><b>Looks</b> let you display actions on the screen, which can be done by adding a <b>say</b> block. The word in the box is called a <b>string</b></p> 	<p>To show a word on the <b>micro:bit</b> display we can use <b>show string</b>, found in the <b>Basic</b> menu</p> 
<p>Blocks that let you edit them have <b>white backgrounds that you can type in</b> for example operators have circular input areas</p> 	<p>Blocks that you can edit usually look like jigsaw pieces and may already have an example in them, for example <b>Math sums have a '0' in the block</b></p> 

# Interface



## Défi 1 : visages

Cette application affiche un visage souriant lorsque le bouton A est enfoncé, un visage triste lorsque le bouton B est enfoncé et un visage endormi lorsque les boutons A et B sont enfoncés.



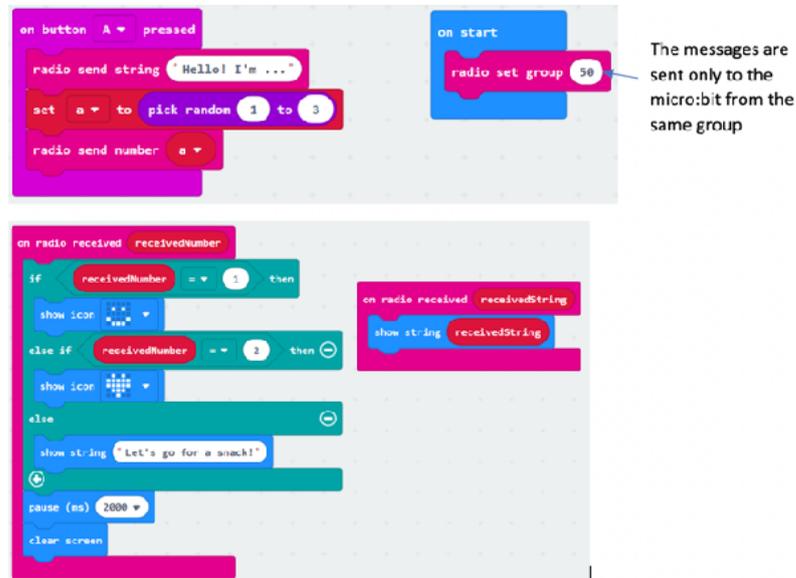
Dans la zone de simulation, il y a des boutons A, B et A + B qui peuvent être utilisés pour simuler ce qui se passe lorsque les boutons micro: bit sont enfoncés.

Pour mettre l'application sur une puce micro: bit, vous devez télécharger le script (sous forme de fichier hex.). Connectez le micro: bit à votre ordinateur avec un câble USB. Localisez le fichier .hex téléchargé et faites-le glisser vers le lecteur MICROBIT.

Cette application est inspirée des tutoriels de <https://makecode.microbit.org/projects/>. Essayez d'autres tutoriels!

## Défi 2 : Chat

L'application suivante permet à deux puces micro: bit ou plus de communiquer entre elles par liaison radio. Les variables *receivedString* et *receivedNumber* sont extraites du bloc de réception radio. L'application doit être téléchargée sur chaque micro: bit.



```

on button A pressed
  radio send string "Hello! I'm ..."
  set a to pick random 1 to 3
  radio send number a

on start
  radio set group 50

on radio received receivedNumber
  if receivedNumber = 1 then
    show icon [cat]
  else if receivedNumber = 2 then
    show icon [dog]
  else
    show string "Let's go for a snack!"
    pause (ms) 2000
    clear screen

on radio received receivedString
  show string receivedString
  
```

The messages are sent only to the micro:bit from the same group

D'autres exemples avec connexion radio sont disponibles sur:

[https:// makecode.microbit.org/projects/radio-games](https://makecode.microbit.org/projects/radio-games)

et

<https://www.instructables.com/id/Radio-Signals-on-Microbit/>.

## Défi 3 : Son et lumière

L'application suivante permet de faire de la musique en faisant varier l'intensité lumineuse sur les capteurs de lumière micro: bit.

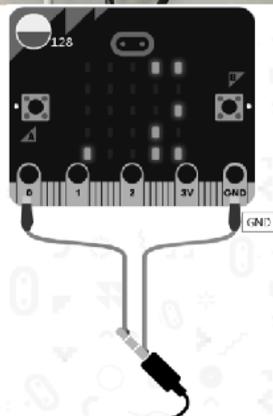
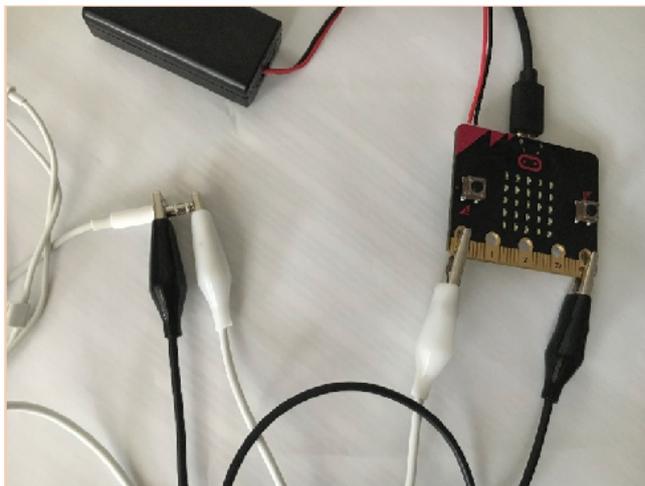
Matériel nécessaire:

1x Micro: bit

1x casque

2 câbles avec pinces crocodile 1x câble USB

Pour commencer, vous devrez connecter le casque au micro: bit comme sur les images suivantes:



```

forever
  rest(ms) 1/2 ▾ beat
  set light ▾ to light level
  if light ▾ < ▾ 25 then
    ring tone (Hz) Middle C
  else if light ▾ < ▾ 50 then ▾
    ring tone (Hz) Middle D
  else if light ▾ < ▾ 100 then ▾
    ring tone (Hz) Middle E
  else if light ▾ < ▾ 150 then ▾
    ring tone (Hz) Middle F
  else if light ▾ < ▾ 180 then ▾
    ring tone (Hz) Middle G
  else ▾
    ring tone (Hz) Middle A
  
```

Définissez une nouvelle variable appelée *lumière* et écrivez le code. Copiez le fichier .hex sur le lecteur micro:bit. Le niveau d'éclairage est une valeur comprise entre 0 (sombre) et 255 (clair). La lumière est mesurée en utilisant différentes LED de l'écran micro:bit.

## Défi 4 : Son et... fruits et légumes

Il s'agit d'une application amusante qui utilise des fruits et légumes pour fermer le circuit et faire de la musique.

Matériel nécessaire:

1x micro: bit

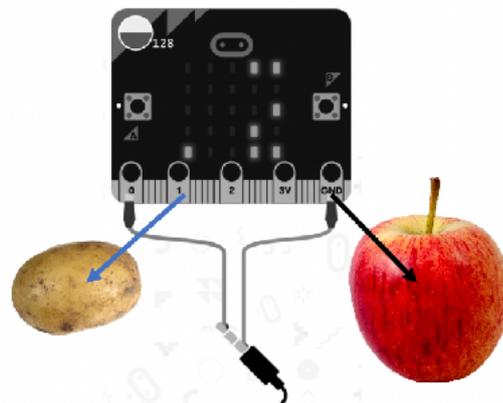
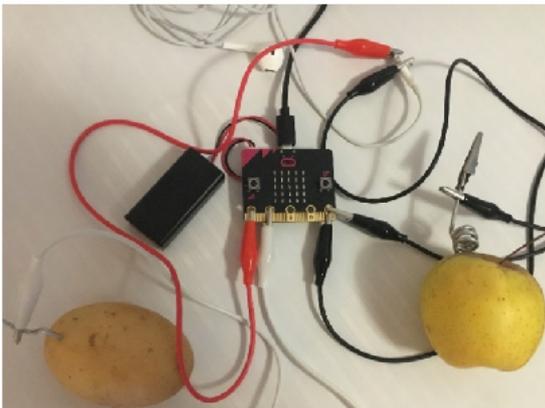
1x casque

4x câbles avec pinces crocodile

1x câble USB

2x fruits ou légumes (pommes de terre, banane, oranges, pommes, etc...)

Faites les connexions suivantes:



Créez une variable nommée son pour stocker une note de musique. Écrivez le code suivant.

```

on pin P1 pressed
  repeat 4 times
    do
      play tone sound for 1/4 beat
      change sound by 25
  set sound to Middle A

on start
  set sound to Middle A
  
```



Enregistrez le fichier .hex et copiez-le sur le lecteur micro:bit. Créez de la musique en tenant le fruit / légume connecté au sol (la pomme dans notre exemple) et en touchant l'autre fruit / légume (connecté à la broche 1).

Vous pouvez connecter un autre fruit / légume à la broche 2 pour créer d'autres sons. Dans ce cas, vous pouvez dupliquer le code de la broche 1, sélectionnez la broche 2 et modifiez, par exemple, la valeur 25 avec -25.

## Défi 5 : Servomoteur

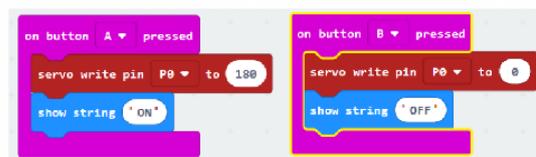
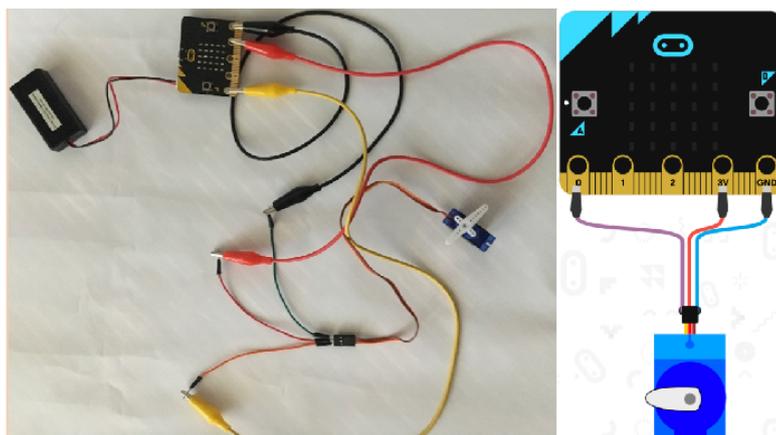
Cette application montre comment connecter un servomoteur à un micro:bit.

Matériel nécessaire:

- 1x micro: bit
- 3 câbles avec pinces crocodile
- 1x câble USB
- 3 fils mâles à mâles
- 1x servomoteur TowerPro SG90

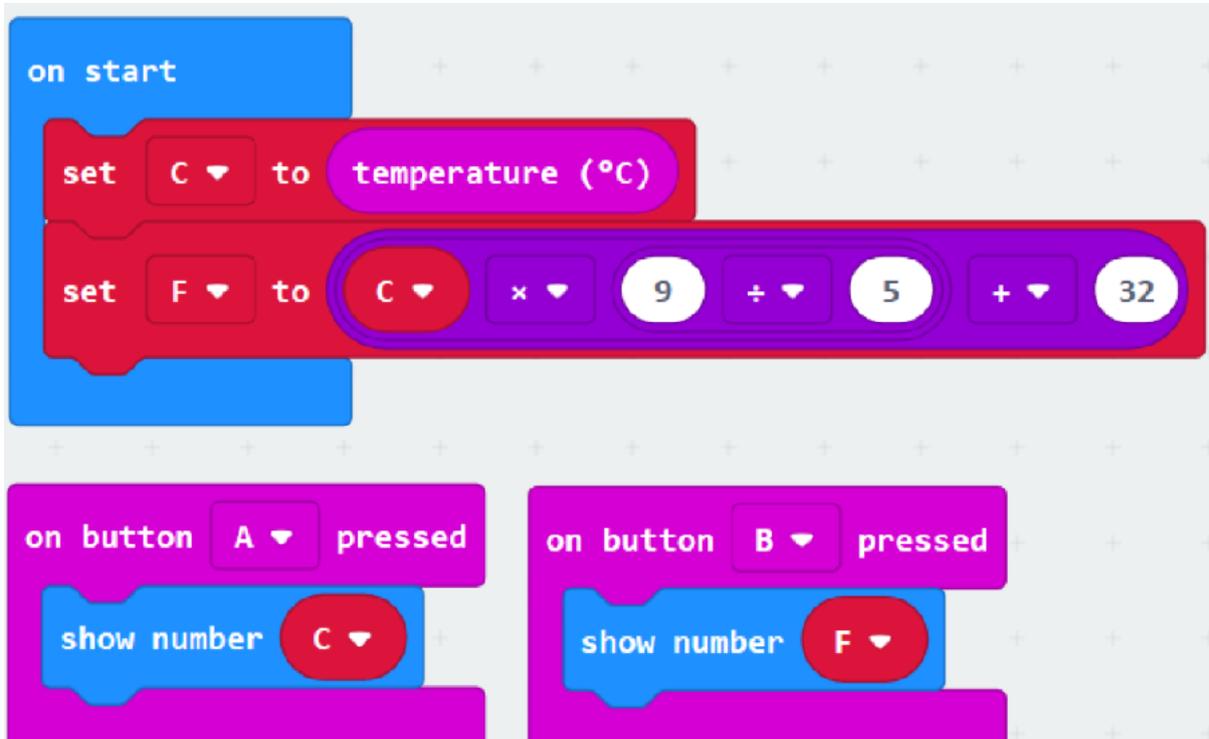
Le servomoteur doit être connecté comme suit:

micro:bit	servomoteur
GND	Fil marron
3V	Fil rouge
P0	Fil orange



## Défi 6 : Thermomètre

Cette application utilise le capteur de température intégré pour afficher la température en degrés Celsius, lorsque le bouton A est enfoncé et en degrés Fahrenheit, le bouton B est enfoncé.



Afin d'obtenir une valeur plus proche de la vraie, vous devez comparer la valeur de micro: bit avec une valeur d'un vrai thermomètre. Ensuite, le programme peut être modifié en soustrayant la différence du nombre affiché par micro: bit.

## Défi 7 : Boussole

L'application suivante affiche la direction cardinale vers laquelle pointe le micro:bit. Après avoir copié le fichier hexadécimal sur le lecteur micro:bit, la puce demandera un étalonnage. Pour cela, vous devrez incliner le micro:bit dans toutes les directions jusqu'à ce que toutes les LED soient allumées. Vous saurez que l'étalonnage a réussi quand un visage heureux est affiché.



```
forever
  set degree to compass heading (°)
  if degree < 45 or degree > 315 then
    show string "N"
  else if degree < 135 then
    show string "E"
  else if degree < 225 then
    show string "S"
  else
    show string "W"
```

The image shows a Scratch script for a compass application. It starts with a 'forever' loop. Inside the loop, the first block is 'set degree to compass heading (°)'. This is followed by an 'if' block with two conditions: 'degree < 45 or degree > 315'. If this condition is true, the script shows the string 'N'. If not, it goes to an 'else if' block with the condition 'degree < 135'. If true, it shows 'E'. If not, it goes to another 'else if' block with the condition 'degree < 225'. If true, it shows 'S'. If none of these conditions are met, it goes to an 'else' block which shows the string 'W'. There are also small minus and plus signs on the right side of the 'else if' and 'else' blocks, indicating they can be collapsed or expanded.

## Plus de défis :

1. Construisez un compteur de pas.
2. Créez votre propre application en gardant à l'esprit le sujet que vous enseignez !

## Ressources

<https://makecode.microbit.org/>

<https://makecode.com/labs>

<https://makecode.microbit.org/projects/>

<https://www.itpro.co.uk/desktop-hardware/26289/13-top-bbc-micro-bit-projects>

<https://www.101computing.net/category/bbc-microbit/>

<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode>

## Références

<https://makecode.microbit.org/projects/>

<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode>



“Que vous souhaitiez découvrir les secrets de l'univers ou que vous souhaitiez simplement poursuivre une carrière au 21e siècle, la programmation informatique de base est une compétence essentielle à apprendre.”

- *Stephen Hawking, physicien théoricien et cosmologiste*