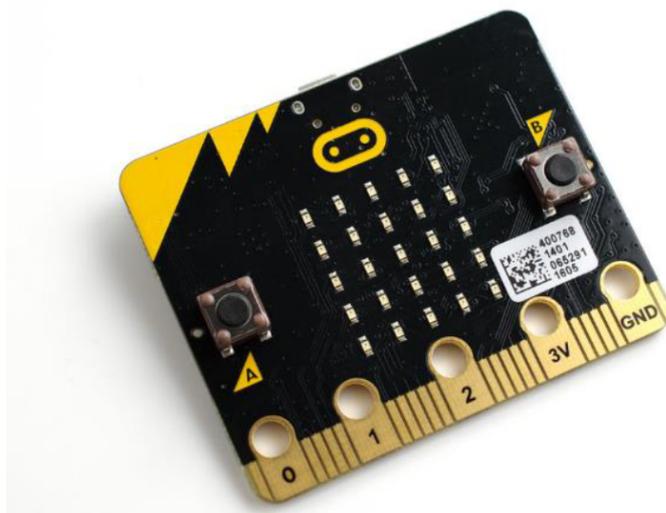


Carte micro-contrôleur **micro:bit** tome 1

Prise en main et programmation par blocs



Commission inter-IREM TICE

contact: c2it@univ-irem.fr

Version du 24 avril 2017

Table des matières

| | | |
|----------|--|-----------|
| 1 | Présentation | 3 |
| 1.1 | Généralités | 3 |
| 1.2 | Fonctionnalités de <code>micro:bit</code> | 4 |
| 1.3 | Où se procurer la carte | 5 |
| 2 | Écrire son programme sur l'interface | 6 |
| 2.1 | Présentation de l'interface de programmation graphique | 6 |
| 2.1.1 | La page principale | 6 |
| 2.1.2 | L'onglet Projets | 6 |
| 2.1.3 | De la programmation par blocs au programme en JavaScript | 7 |
| 2.2 | Les menus | 7 |
| 2.3 | Le menu Variables | 10 |
| 2.4 | Conditionnelles imbriquées et « si...ou si...sinon... » (if...elif...else) | 10 |
| 2.5 | Le menu Avancé - Texte | 11 |
| 3 | Tutoriel de prise en main | 11 |
| 4 | Exemples de projets tirés ou adaptés du site officiel | 11 |
| 4.1 | Thermomètre | 11 |
| 4.2 | Boussole | 12 |
| 4.3 | Les « Cœurs clignotants » (Flashing hearts) | 12 |
| 4.4 | Smiley piloté à distance | 13 |
| 5 | Projets de groupes IREM | 13 |
| 5.1 | Programmer un morceau de musique | 13 |
| 5.2 | Le jeu de la devinette | 13 |
| 5.3 | Lancer de dés | 14 |
| 5.3.1 | Les dés à 6 faces classiques | 15 |
| 5.3.2 | Les dés de Sicherman | 16 |
| 5.3.3 | Les dés d'Efron | 17 |
| 6 | Transférer un programme sur <code>micro:bit</code> depuis un PC via un port USB | 18 |
| 7 | Transférer un programme sur <code>micro:bit</code> depuis un appareil Android | 18 |
| 7.1 | Appairer la carte avec un appareil Bluetooth | 18 |
| 7.2 | Connecter la carte avec l'appareil Bluetooth appairé | 19 |

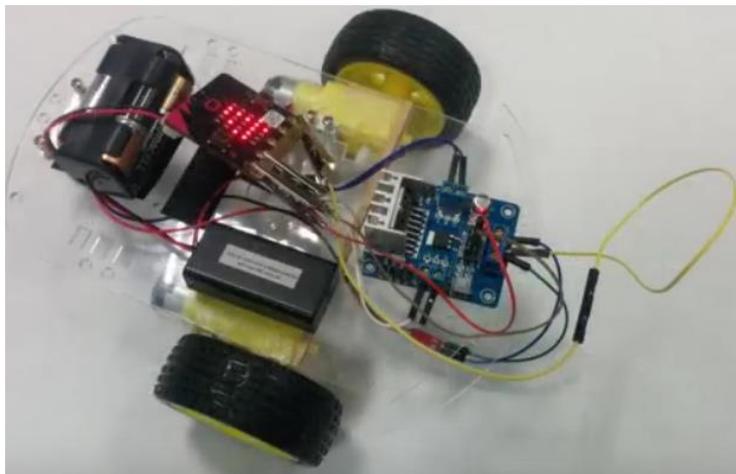
1 Présentation

1.1 Généralités

BBC micro:bit (que nous appellerons dans la suite micro:bit) est une carte micro-contrôleur qui a été conçue par la BBC, accompagnée de 29 partenaires dont Samsung, ARM et Microsoft, dans un but purement pédagogique. Elle permet à des élèves de niveau collège d'aborder la robotique avec un langage graphique simple analogue à Scratch tout en autorisant la conception de systèmes automatisés en cours de technologie ou de sciences. Elle mesure seulement 52×42 mm et est conçue pour les jeux et les applications connectées. Elle est programmable depuis les PC, les appareils Android et IOS.

Concrètement de quoi s'agit-il? On parle ici de micro-contrôleur, à savoir une carte électronique programmable pour interagir avec le monde réel. C'est une version accessible de l'électronique que tout un chacun manipule au quotidien sans se poser de question, par exemple les dispositifs de domotique qui permettent de gérer à distance le chauffage, la sécurité, l'arrosage du géranium... Ou bien plus simplement la bouilloire programmable au degré Celsius près, la guirlande du sapin qui clignote au rythme de « Vive le vent ». Ce micro-contrôleur permet d'élaborer par exemple un podomètre, un doudou sensoriel, un sismographe rudimentaire... Bien entendu de nombreux exemples de projets existent, qu'ils soient issus des émissions BBC ou de la communauté éducative. Sur le site officiel on trouve des idées, des tutoriels, des leçons (<http://microbit.org/fr/ideas/>) comme par exemple : une alarme de trousse, un compteur de frappe (au baseball) ou encore des leçons sur l'accélération.

Exemple de projet : Voiture radio-commandée



Pour en savoir plus :

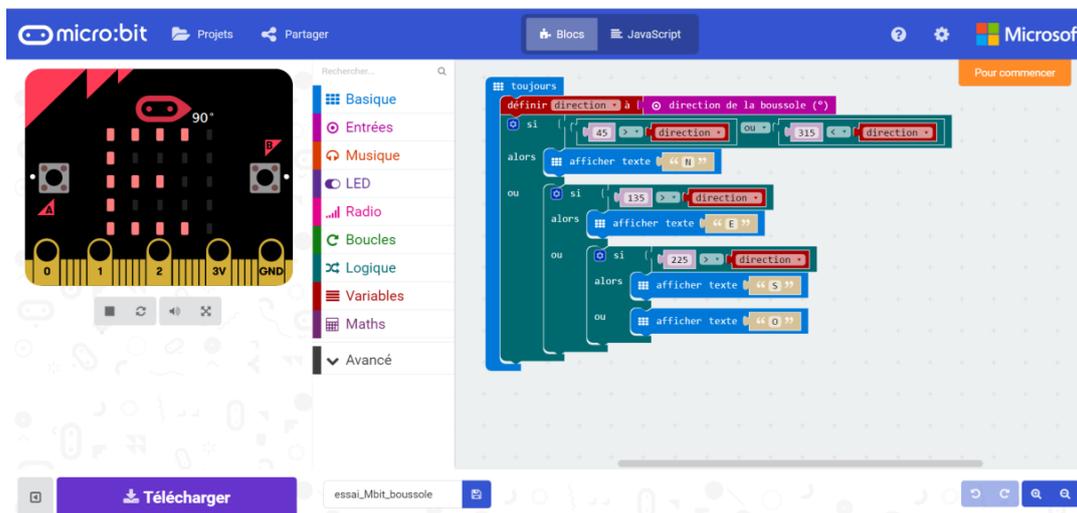
- vidéo du projet <https://www.youtube.com/watch?v=nQ2k-mjp5UY>
- la carte de pilotage des moteurs (pour protéger le micro:bit et pour fournir la puissance nécessaire, seul des petits servomoteurs peuvent être pilotés directement) <https://www.youtube.com/watch?v=S1nCNtucgpA>
- le Kitronic Inventors Kit <https://www.youtube.com/watch?v=1mdzM74XyHw>
- les possibilités liées au Bluetooth https://www.youtube.com/watch?v=aep_GVowKfs

Des ressources pour l'enseignement sont proposées sur le site <http://microbit.org/fr/teach/>.

Ce micro-contrôleur peut être programmé via une interface en ligne :

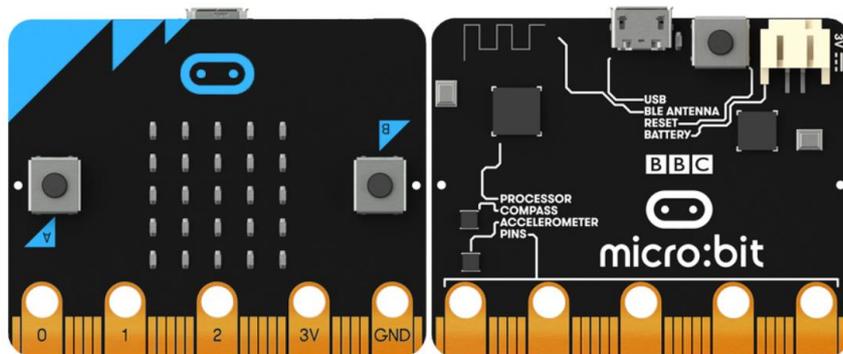
- en langage Python <http://python.microbit.org/editor.html> : voir le tutoriel à l'adresse <https://microbit-micropython.readthedocs.io/en/latest/>
- En JavaScript <https://pxt.microbit.org/?lang=fr>.

L'interface de programmation par blocs de la version JavaScript permet une utilisation pédagogique au collège. Nous nous concentrerons sur cette interface dans ce document. Des projets sont proposés sur le site.



Sur la gauche de la fenêtre, un émulateur micro:bit permet de simuler l'exécution des programmes avant de les transférer sur une carte réelle. Ceci simplifie les usages en classe et à la maison puisque le matériel n'est pas nécessaire pour tester les programmes.

1.2 Fonctionnalités de micro:bit



La carte contient un certain nombre de capteurs :

- Un bouton de réinitialisation, libellé « R » pour « Reset »
- Deux boutons programmables, libellés « A » et « B » sur la carte
- Un accéléromètre pour détecter les changements de vitesse de l'appareil (qui permet par exemple de détecter des actions comme secouer, pencher et la chute libre)
- Une boussole magnétique 3D pour détecter les champs magnétiques
- Un capteur de température (sur le processeur)
- Un capteur de luminosité lié aux diodes
- Une connectique bluetooth 4.0 basse énergie/2.4 GHz maître/esclave

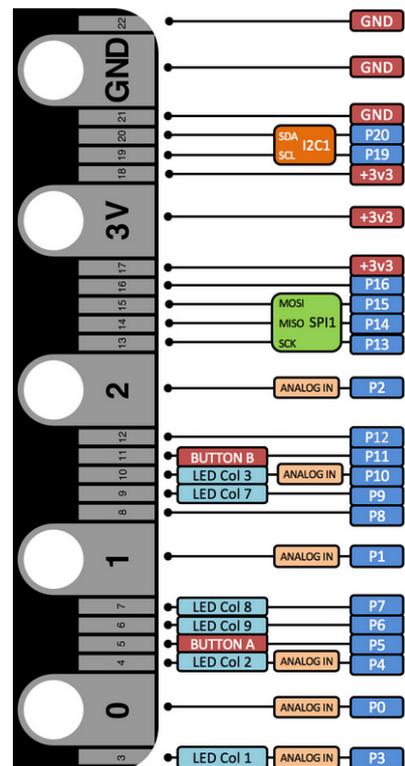
Elle contient également :

- afficheur digital carré de 25 LED (5 × 5) rouges programmables pouvant servir d'affichage, notamment pour des motifs animés, du texte alphanumérique déroulant

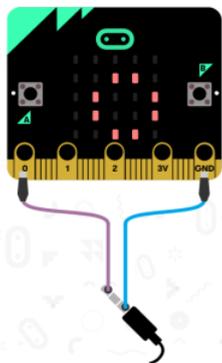
25 connecteurs externes permettant de multiples entrées-sorties (thermomètre, écouteurs...) :

- 5 ports d'entrée-sortie en forme d'anneau. Chacune des 5 E/S est programmable, pour être traitée soit en analogique, soit en numérique. Les anneaux sont compatibles avec des prises crocodile ou des fiches banane 4mm ;
- 1 port Masse référencé « GND », libellé « GND » sur la carte
- 1 port Puissance référencé « PWR » et fournissant 3 Volts, éventuellement pour alimenter un autre appareil, libellé « 3V » sur la carte
- 3 autres ports, référencés « Pins » P0 à P2, libellés « 0 », « 1 », « 1 » sur la carte ;
- Un connecteur latéral à 20 broches, à connexion standard, référencés « Pins » P3 à P22. Ceci permet de connecter à un appareil comme un Arduino, Galileo, Kano ou encore un Raspberry Pi

Voir <http://microbit.org/fr/hardware/pins/>



Par exemple, pour connecter des enceintes ou une paire d'écouteurs à la carte, il suffit de deux câbles avec pinces crocodiles et de suivre les instructions données par l'émulateur :



La carte peut être alimentée par l'entrée batterie ou par le port micro-USB femelle.

Le port USB permet également les transferts de programme depuis les PC.

Pour en savoir plus : <http://microbit.org/fr/hardware/>.

1.3 Où se procurer la carte

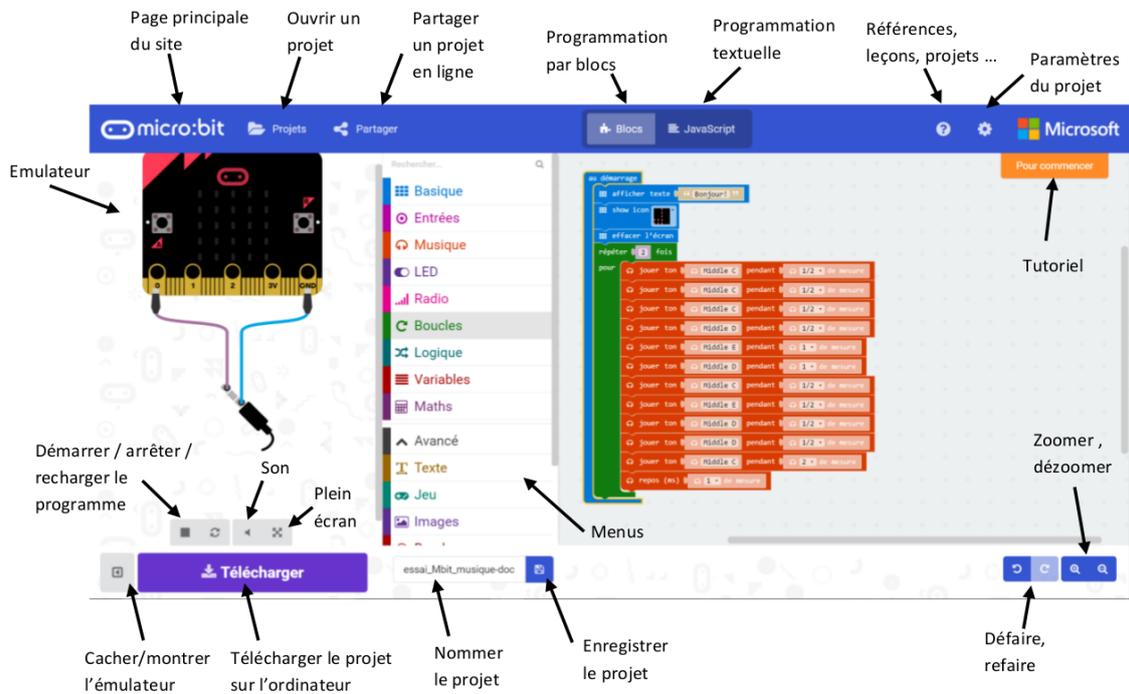
Un kit comprenant : carte micro:bit + câble USB + bloc piles + piles coûte environ 25€ (200€ pour 10 kits). On peut se procurer la carte et accessoires de connexion sur le site français Kubii. La carte et ses accessoires, extensions et kit de démarrage sont en vente sur le site britannique Kitronik.

- <https://www.kubii.fr/fr/146-microbit>
- <https://www.kitronik.co.uk/microbit.html> et <https://www.kitronik.co.uk/microbit/bbc-micro-bit-accessories.html>

2 Écrire son programme sur l'interface

2.1 Présentation de l'interface de programmation graphique

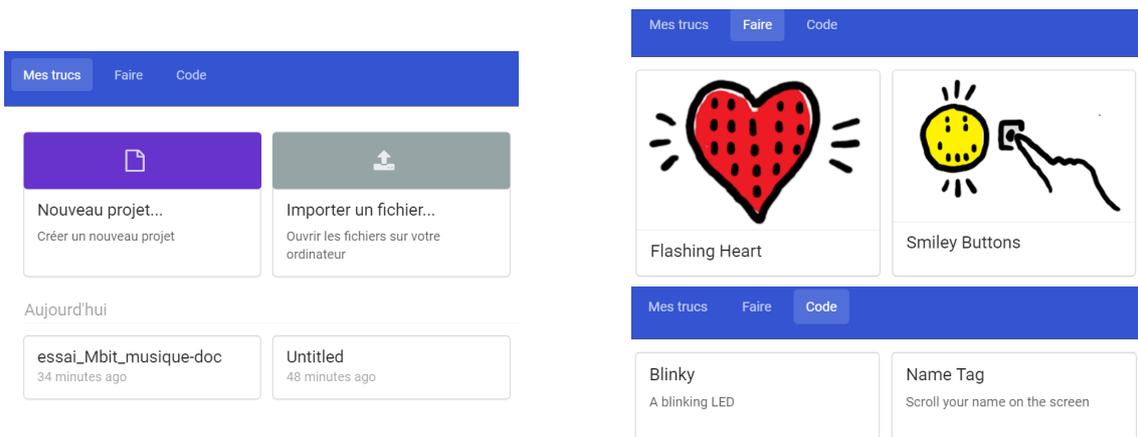
2.1.1 La page principale



Comme en Scratch, il suffit de glisser les blocs et de les assembler comme un puzzle.

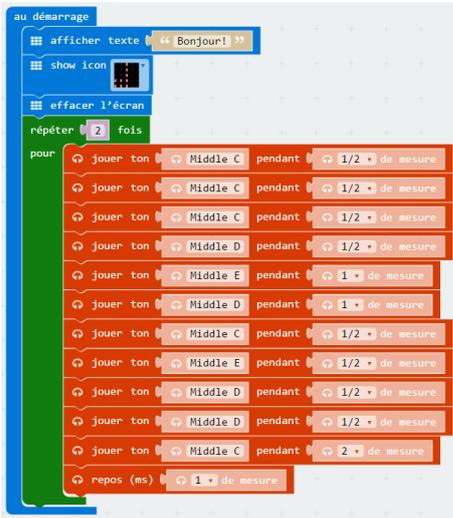
2.1.2 L'onglet Projets

On y retrouve ses propres scripts dans l'onglet « Mes trucs », des mini projets dans l'onglet « Faire » et des scripts à tester dans l'onglet « Code » :



2.1.3 De la programmation par blocs au programme en JavaScript

Pour visualiser le programme en JavaScript d'un programme écrit avec des blocs il suffit de cliquer sur **JavaScript** pour convertir le script :



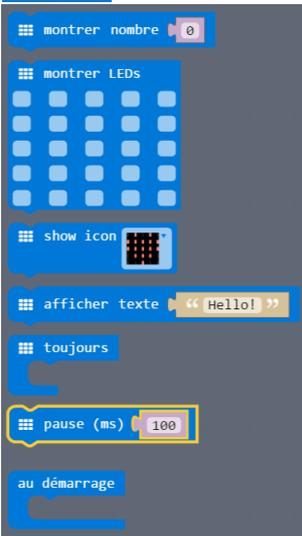
```

1 basic.showString("Bonjour!")
2 basic.showIcon(IconNames.QuarterNote)
3 basic.clearScreen()
4 for (let i = 0; i < 2; i++) {
5   music.playTone(262, music.beat(BeatFraction.Half))
6   music.playTone(262, music.beat(BeatFraction.Half))
7   music.playTone(262, music.beat(BeatFraction.Half))
8   music.playTone(294, music.beat(BeatFraction.Half))
9   music.playTone(330, music.beat(BeatFraction.Whole))
10  music.playTone(294, music.beat(BeatFraction.Whole))
11  music.playTone(262, music.beat(BeatFraction.Half))
12  music.playTone(330, music.beat(BeatFraction.Half))
13  music.playTone(294, music.beat(BeatFraction.Half))
14  music.playTone(294, music.beat(BeatFraction.Half))
15  music.playTone(262, music.beat(BeatFraction.Double))
16 }
17
18

```

2.2 Les menus

Basique



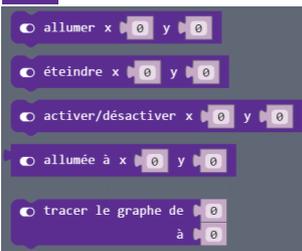
Entrée



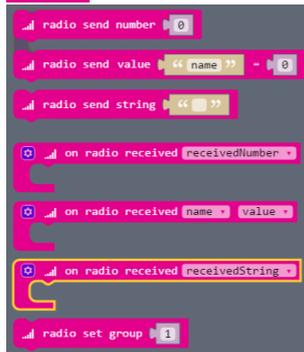
Musique



LED



Radio

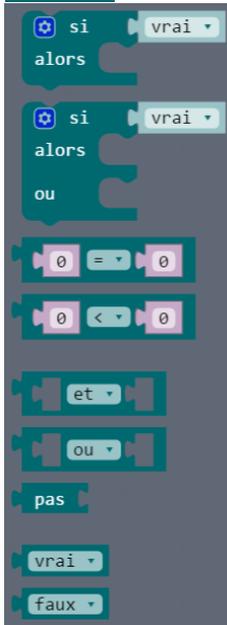


Boucle



L'utilisation des blocs du menu radio fait apparaître deux cartes sur l'émulateur.

Logique



Variable

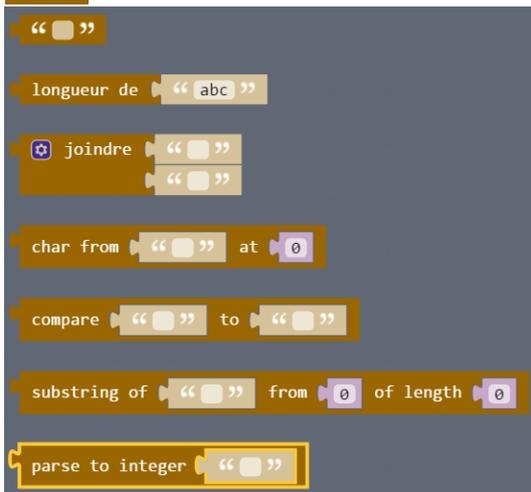


Maths

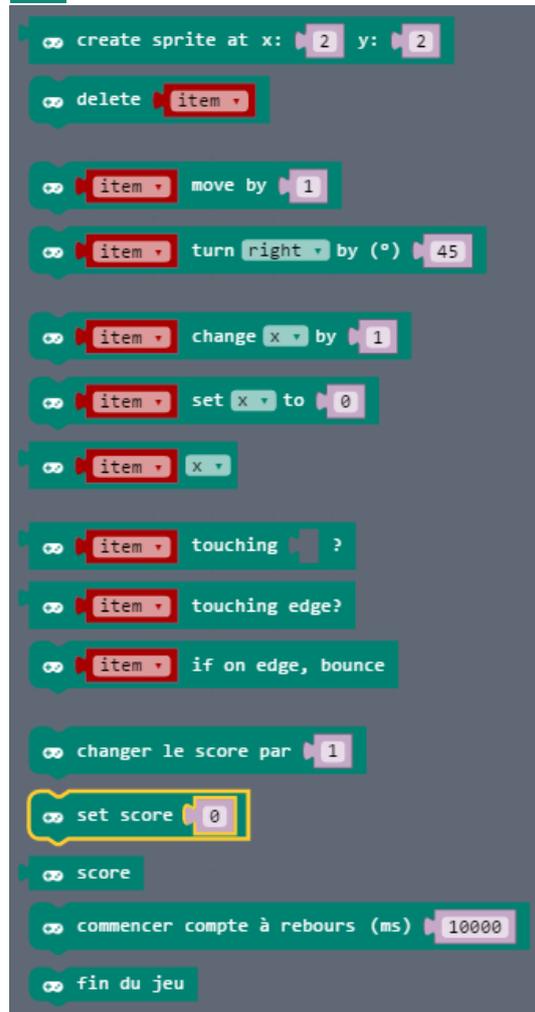


Les menus **Avancé** :

Texte



Jeu



Image

show image **item** at offset **0**

scroll image **item** with offset **1** and interval (ms) **200**

créer image

créer grande image

arrow image **North**

North

icon image

Communication série

série écrire ligne " " "

série écrire nombre **0**

série écrire valeur "x" = **0**

série écrire texte " " "

série lire ligne

série lire jusqu'à **new line**

serial on data received **new line**

serial read string

série rediriger vers

TX **P0**

RX **P1**

au débit en baudes **14400**

Broche

lire en numérique broche **P0**

écrire en numérique la broche **P0** à **0**

lire la broche analogique **P0**

analog write pin **P0** to **1023**

analog set period pin **P0** to (μ s) **20000**

map

from low **0**

from high **1023**

to low **0**

to high **4**

servo écrire broche **P0** à **180**

servo set pulse pin **P0** to (μ s) **1500**

lorsque la broche **P0** est pulsée **high**

durée de l'impulsion (μ s)

pulse in (μ s) pin **P0** pulsed **high**

i2c lire nombre à l'adresse **0** de format **Int8LE**

i2c écrire nombre à l'adresse **0** avec valeur **0** de format **Int8LE**

spi écrire **0**

set pin **P0** to emit **edge** events

analog pitch **0** for (ms) **0**

set pull pin **P0** to **up**

analog set pitch pin **P0**

Contrôle

exécuter en arrière-plan

remise à zéro

attendre (μ s) **4**

raise event

from source **MICROBIT_ID_BUTTON_A**

with value **MICROBIT_EVT_ANY**

on event

from **MICROBIT_ID_BUTTON_A**

with value **MICROBIT_EVT_ANY**

event timestamp

event value

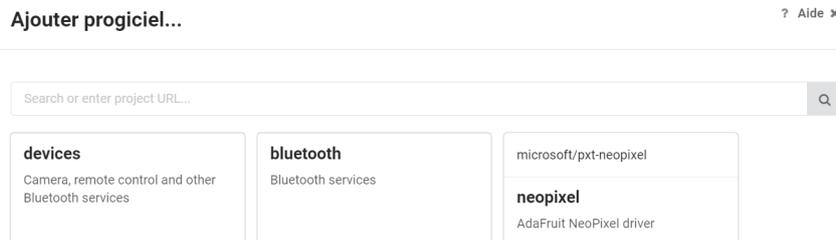
MICROBIT_ID_BUTTON_A

MICROBIT_EVT_ANY

device name

device serial number

Via le menu **Avancé**, il est également possible d'ajouter des progiciels c'est-à-dire des paquets contenant des fonctions et des types pouvant être utilisés avec la carte :



Pour en savoir plus : <https://pxt.microbit.org/packages>.

2.3 Le menu Variables

Les fonctions du menu Variables ne sont pas très explicites ou bien traduites :

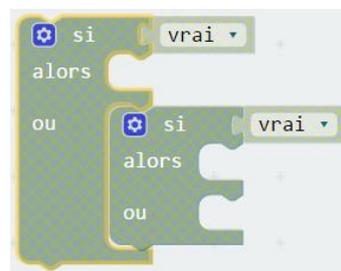
| Version française | Version anglaise | Signification |
|---------------------|------------------|---|
| définir item à 0 | set item to 0 | Mettre la variable item à la valeur indiquée |
| changer item pour 1 | change item by 1 | Ajouter à la variable item la valeur indiquée |

2.4 Conditionnelles imbriquées et « si...ou si...sinon... » (if...elif...else)

Dans l'interface en français de programmation par blocs, la structure de contrôle « si...alors...sinon... » (« if...then...else... ») s'appelle :

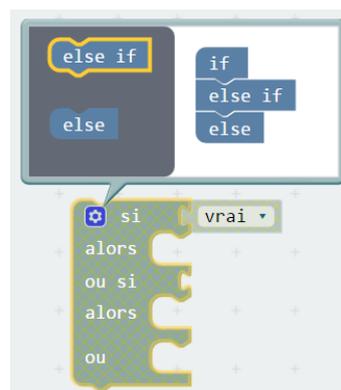


Imbriquer des instructions conditionnelles peut être long et fastidieux. Comme en langage Python, l'interface de micro:bit permet de remplacer une imbrication de deux



« si...ou si...ou... » (« if...elif...else »).

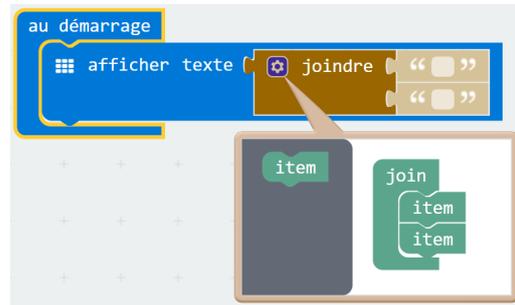
Sur le bloc que l'on trouve dans le menu logique, se trouve une roue paramètre en haut à gauche. Une fenêtre apparaît et disparaît quand on clique sur la roue. Dans cette fenêtre on peut concevoir la structure conditionnelle souhaitée en glissant les blocs depuis la partie sur fond gris vers la partie sur fond blanc en les assemblant.



2.5 Le menu Avancé - Texte

Dans ce menu on trouve une fonction permettant de concaténer des chaînes de caractères. Comme pour les `si`, `alors`, `ou`, la roue permet de paramétrer le bloc.

En glissant des blocs `item` dans le bloc `joindre`, on définit le nombre de chaînes à concaténer.



3 Tutoriel de prise en main

D'après le site <https://pxt.microbit.org/?lang=fr>.

En allant sur cette page et en cliquant sur `Pour commencer` vous accédez à un tutoriel en anglais. En voici une traduction.

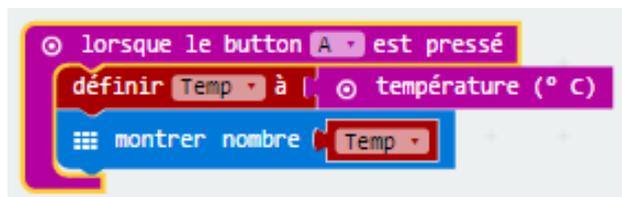
1. Bienvenue! Placez le bloc `afficher texte` dans le bloc `au démarrage` pour pour faire défiler votre nom lors du démarrage du programme.
2. Cliquez sur `Télécharger` pour transférer votre code dans votre micro:bit!
3. Le texte a cessé de défiler. Placez le bloc `afficher texte` dans le bloc `Lorsque le bouton A est pressé` pour faire défiler votre nom quand le bouton A est pressé.
4. Cliquez sur `Télécharger` pour transférer votre code dans votre micro:bit et pressez le bouton A pour faire défiler votre nom.
5. Placez des blocs pour afficher un Smiley quand le bouton B est pressé. Transférez votre code dans votre micro:bit et pressez le bouton B pour faire apparaître le Smiley.
6. Placez les blocs `montrer nombre` et `choisir au hasard de 0 à` dans le bloc `lorsque secouer` pour fabriquer un dé. Transférez votre code dans votre micro:bit et secouez la carte pour simuler des lancers de dés.
7. Cliquez sur terminer pour retourner à l'interface de programmation.

4 Exemples de projets tirés ou adaptés du site officiel

Les exemples suivant sont tirés du site <http://microbit.org/fr>.

4.1 Thermomètre

Programmer la carte pour qu'elle affiche la température à l'aide de son capteur interne quand on presse le bouton A.

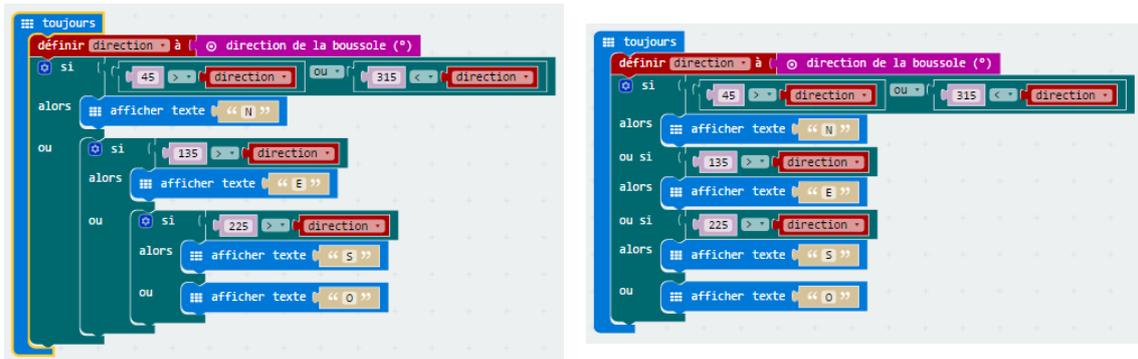


4.2 Boussole

Programmer la carte pour qu'elle affiche l'orientation sous la forme d'une lettre sur l'afficheur digital (N, S, E, O) à tout instant.

- Le capteur renvoie 0° ou 360° quand la boussole est orientée au Nord.
- Il renvoie 90° quand la boussole est orientée à l'est.
- Il renvoie 180° quand la boussole est orientée au Sud.
- Il renvoie 270° quand la boussole est orientée à l'ouest.

Penser à utiliser les structures conditionnelles simplifiées.



Modifier le programme pour qu'il affiche aussi des orientations intermédiaires (SE, SO, NE, NO).

Pour comprendre en quoi la structure simplifiée facilita la modification, voir

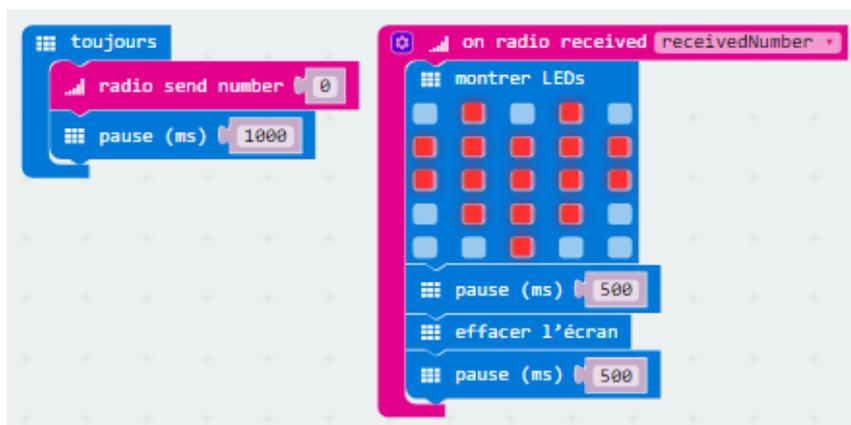
<https://pxt.microbit.org/projects/compass> et

<https://www.microbit.org/en/2017-03-07-javascript-block-resources/>.

4.3 Les « Cœurs clignotants » (Flashing hearts)

Programmer deux cartes qui quand elles envoient/reçoivent un signal radio, affichent des cœurs qui battent alternativement.

Cet exercice permet de travailler la liaison radio entre deux cartes et peut être proposé avec le guidage suivant <https://pxt.microbit.org/projects/flashing-heart> et <https://www.microbit.org/en/2017-03-07-javascript-block-resources/>.

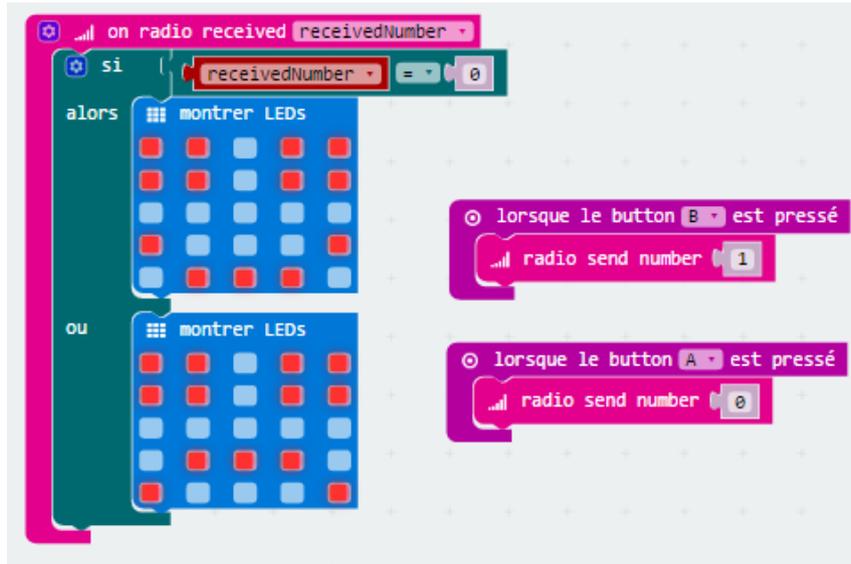


4.4 Smiley piloté à distance

Avec deux cartes, faire un programme qui :

- Quand le bouton A est pressé sur la carte 1 affiche un smiley content sur l'afficheur numérique de la carte 2.
- Quand le bouton B est pressé sur la carte 1 affiche un smiley mécontent sur l'afficheur numérique de la carte 2.

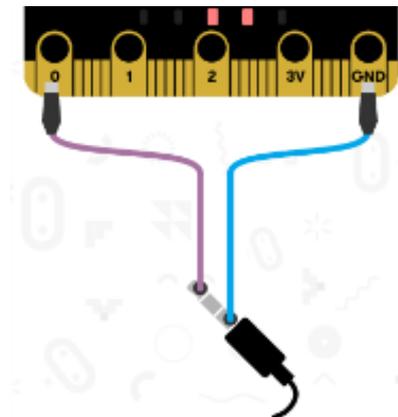
Voir <https://www.microbit.org/en/2017-03-07-javascript-block-resources/> et aussi <https://pxt.microbit.org/projects/smiley-buttons>.



5 Projets de groupes IREM

5.1 Programmer un morceau de musique

Programmer le morceau de musique de votre choix et connecter la carte à une paire d'écouteurs à l'aide de pinces crocodiles.



5.2 Le jeu de la devinette

L'utilisateur choisit un nombre entre 0 et 1000. Programmer la carte pour qu'elle devine le nombre le plus rapidement possible. À chaque proposition de la carte sur l'afficheur numérique, l'utilisateur

presse simultanément sur les boutons A et B si le résultat est trouvé, il presse sur A si le nombre à trouver est plus petit que la suggestion et sur B s'il est plus grand.

À la fin, le nombre d'essais apparaît sur l'afficheur et une petite musique annonce la fin du jeu. Le jeu recommence automatiquement.

```

au démarrage
  afficher texte " Nombre entre 0 et 1000 ? "
  définir Xmin à 0
  définir Xmax à 1000
  définir nb coups à 1
  définir S à Xmax
  montrer nombre S

lorsque le bouton A est pressé
  définir Xmax à S
  définir S à (Xmin + Xmax) / 2
  montrer nombre S
  changer nb coups pour 1

lorsque le bouton B est pressé
  définir Xmin à S
  définir S à (Xmin + Xmax) / 2
  montrer nombre S
  changer nb coups pour 1

lorsque le bouton A+B est pressé
  start melody "dadadum" repeating once
  afficher texte " Bravo! "
  afficher texte joindre nb coups " coups "
  
```

Ce jeu permet de travailler la méthode de dichotomie et peut faire suite à une activité débranchée sur le sujet.

5.3 Lancer de dés

Sur le thème des lancer de dés, plusieurs projets peuvent être proposés en lien avec les probabilités.

5.3.1 Les dés à 6 faces classiques

- Lancer d'un dé à 6 face classique quand on secoue la carte avec visualisation de la face obtenue sur l'afficheur numérique. Ce programme se fait sans boucle.

```

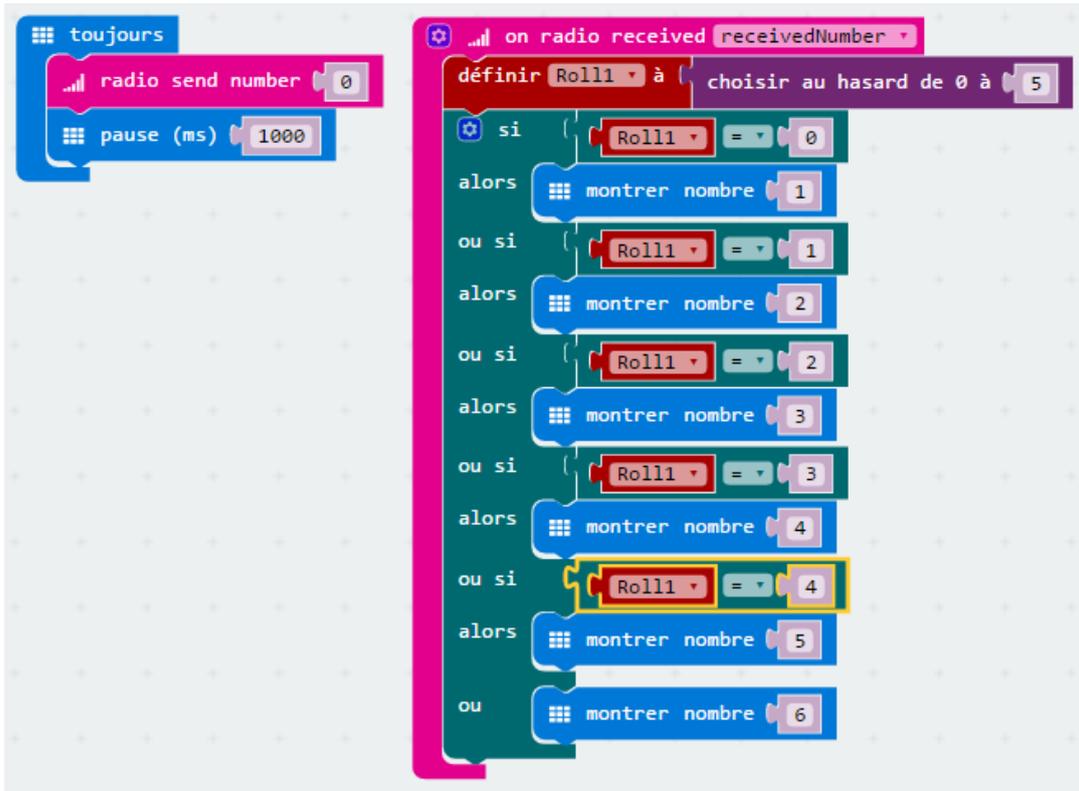
    lorsque secouer
    définir Roll à choisir au hasard de 0 à 5
    si Roll = 0
    alors montrer LEDs
    ou si Roll = 1
    alors montrer LEDs
    ou si Roll = 2
    alors montrer LEDs
    ou si Roll = 3
    alors montrer LEDs
    ou si Roll = 4
    alors montrer LEDs
    ou Roll = 5
    alors montrer LEDs
  
```

- N lancers successifs d'un dé à 6 faces classique quand on clique sur le bouton A avec affichage de chaque nombre obtenu sur l'afficheur numérique. Ce programme se fait avec boucle.

```

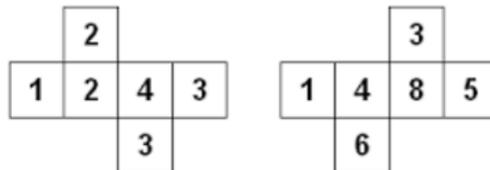
    lorsque le bouton A est pressé
    répéter 10 fois
    pour
    montrer nombre 1 + choisir au hasard de 0 à 5
    pause (ms) 100
    afficher texte " "
    pause (ms) 100
  
```

- Utiliser le menu radio pour faire des lancers de 2 dés à 6 faces normaux.

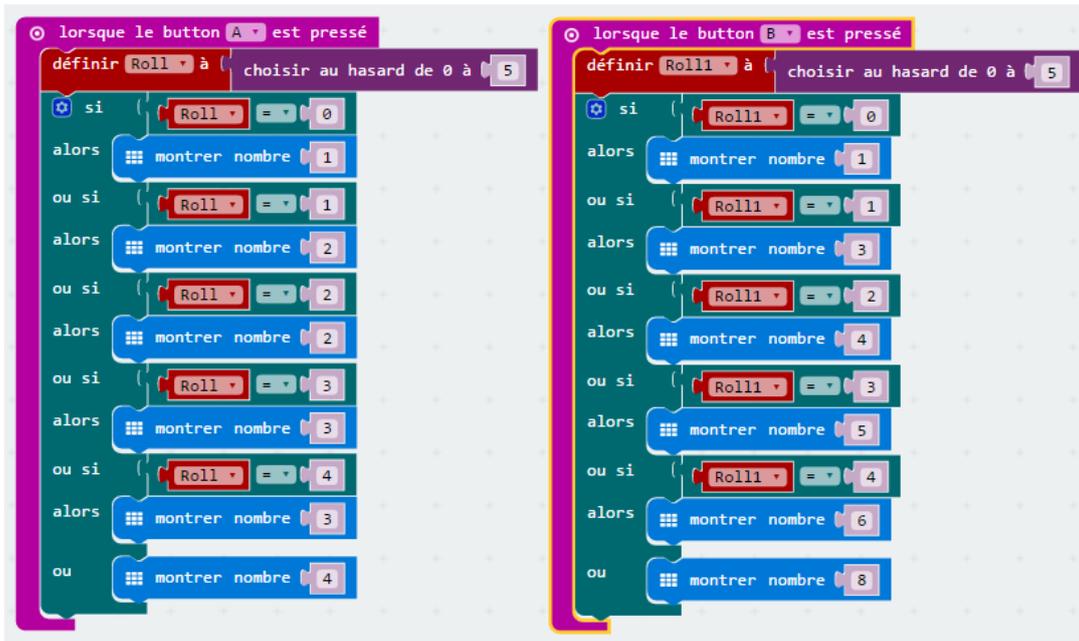


5.3.2 Les dés de Sicherman

- Modifier le ou un des codes précédents pour changer les dés classiques en dés de Sicherman : Les faces des dés de Sicherman sont numérotées 1, 2, 2, 3, 3, 4 pour l'un et 1, 3, 4, 5, 6, 8 pour l'autre. Utiliser les boutons A et B pour simuler le lancer de chacun des dés de Sicherman.

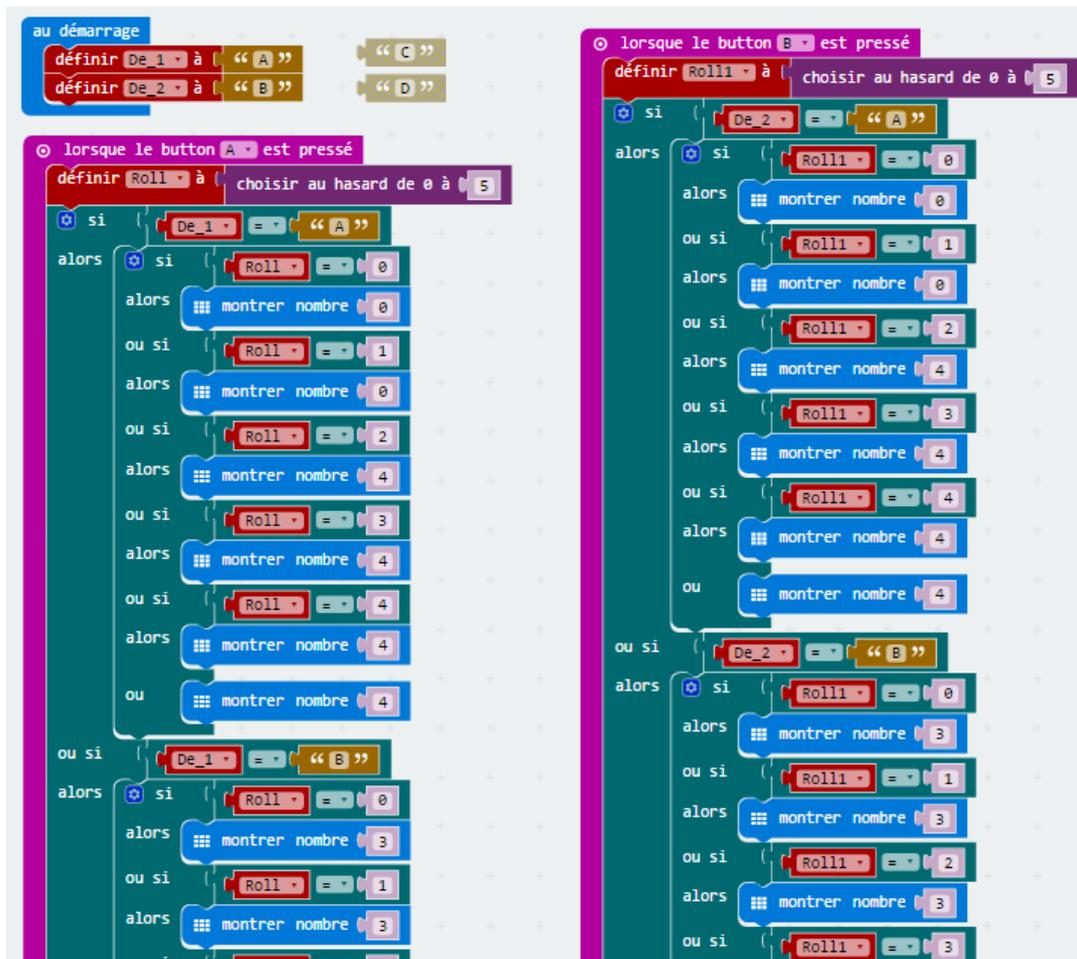


- Jouer au jeu suivant :
 - Le joueur 1 a 2 dés classiques
 - Le joueur 2 a une paire de dés de Sicherman
 - Chaque joueur lance ses deux dés et calcule la somme obtenue. Celui qui a la somme la plus grande marque un point. Et on recommence.
- Programmer le jeu. Ce jeu est-il équitable? Expliquer pourquoi.
- Réponse : La probabilité d'obtenir un résultat particulier avec les dés de Sicherman est la même qu'avec des dés ordinaires...et ce sont les seuls dés avec des nombres strictement positifs possédant cette propriété (https://fr.wikipedia.org/wiki/Dés_de_Sicherman).



5.3.3 Les dés d'Efron

- Programmer le jeu des dés d'Efron (https://fr.wikipedia.org/wiki/Dés_non_transitifs). Les dés d'Efron sont un jeu de quatre dés inventés par Bradley Efron. Les quatre dés A, B, C, D portent les numéros suivants sur leurs six faces :
 - A : 4, 4, 4, 4, 0, 0 ;
 - B : 3, 3, 3, 3, 3, 3 ;
 - C : 6, 6, 2, 2, 2, 2 ;
 - D : 5, 5, 5, 1, 1, 1.
 - Chaque joueur choisi un des 4 dés (ils ne peuvent avoir le même). Un joueur marque un point quand lors d'un lancer le résultat qu'il a obtenu est le plus grand. Quelle est la probabilité qu'un dé en batte un autre ? Quelle stratégie doit-on adopter pour gagner si on choisit son dé en premier ? Si on choisit son dé en deuxième ? Expliquer pourquoi.
 - Indications extraites de Wikipédia :
 - La probabilité que A batte B, B batte C, C batte D et D batte A est égale à $2/3$.
 - Les autres probabilités varient suivant les dés :
 - la probabilité que A batte C est $4/9$;
 - la probabilité que B batte D est égale à $1/2$;
 - la probabilité que C batte A vaut $5/9$;
 - la probabilité que D batte B est $1/2$.
 - En revanche, la probabilité qu'un dé en batte un autre pris au hasard parmi les trois restants n'est pas égale suivant les dés :
 - dans le cas de A, elle vaut $13/27$;
 - pour B, $1/2$;
 - pour C, $14/27$;
 - pour D, $1/2$.
- Globalement, le meilleur dé pour gagner un jeu totalement aléatoire est donc C, qui gagne dans près de 52 % des cas.



6 Transférer un programme sur micro:bit depuis un PC via un port USB

- Connecter la carte au PC via le port USB.
- Télécharger sur le PC le programme fait sur l'interface (fichier .hex)
- À l'aide du gestionnaire de fichiers, copier sur le micro:bit le fichier téléchargé.
Attention, si la carte semble se comporter au début comme une clé USB, une fois copié, le fichier n'apparaît plus dans le dossier de la carte mais le programme a bien été transféré dans le processeur.

7 Transférer un programme sur micro:bit depuis un appareil Android

7.1 Appairer la carte avec un appareil Bluetooth

Avant d'appairer un appareil Android à la carte, il est conseillé d'y avoir au préalable mis un programme via l'entrée USB depuis un PC. Ceci permet de remettre à jour le logiciel de la carte.

- Installer l'application micro:bit de Samsung depuis le Play Store <https://play.google.com/store/apps/details?id=com.samsung.microbit&hl=fr>
- Activer le Bluetooth de l'appareil Android
- Lancer l'application micro:bit
- Sélectionner connexion
- Appairer une nouvelle carte micro:bit et suivre les instructions.

- Appuyer simultanément sur les boutons A et B et, sans les relâcher, sur le bouton Reset qui se trouve de l'autre côté de la carte pendant une seconde. Garder A et B pressé jusqu'à ce que la phrase « PAIRING MODE! » s'affiche.
- Un motif s'affiche sur l'afficheur digital de la carte. Reporter ce motif sur l'application. Sur chaque colonne, il suffit de taper le carré à allumer le plus haut pour que lui et les carrés qui sont en dessous s'allument.
- Dans l'application, taper sur suivant et quand un numéro de pin est demandé, presser le bouton A de la carte (indiqué par ←) et entrer dans l'application les numéros qui défilent sur la carte. Ces numéros tournent en boucle pendant 30 secondes.
- Quand la carte est appairée, le symbole coche s'affiche sur la carte et l'application indique qu'il faut presser le bouton Reset. Presser le bouton.

7.2 Connecter la carte avec l'appareil Bluetooth appairé

Attention, une fois la carte et l'appareil appairés, les tentatives de connexion via l'application en cliquant sur connecter à un appareil déjà appairé ou en passant directement par le Flashage d'un programme peuvent conduire à un message d'erreur : « Connexion unsuccessful » ou « Flashing unsuccessful GATT ERROR ». . . Il est alors suggéré d'appuyer sur le bouton Reset de la carte et de réessayer, ce qui ne semble pas régler le problème.

Il est possible de palier au problème avec la méthode suivante :

- Appuyer simultanément sur les boutons A et B et, sans les relâcher, sur le bouton reset qui se trouve de l'autre côté de la carte pendant une seconde. Garder A et B pressés jusqu'à ce que le mot « PAIRING » s'affiche.
- Un motif s'affiche sur l'afficheur digital de la carte.
- Sélectionner connexion
- Connecter la carte déjà appairée. Sur l'application, le connecteur doit devenir vert.
- Aller dans le menu Flash, choisir le programme et le flasher.
- Attention, ne pas agir sur la carte tant que le transfert du programme n'est pas terminé.