



Variables et conditionnelles

bloc - 1 période

Programmation - Variable - Conditions - Découverte - Accéléromètre

Compétences

Mathématique

- Interpréter un tableau de nombres, un graphique, un diagramme
- Vérifier le résultat d'une opération

Éducation par la technologie

- Imaginer des dispositifs expérimentaux simples et prendre des initiatives
- Dans le cadre d'une énigme, agencer les indices en vue de formuler au moins une question, une supposition ou une hypothèse
- Élaborer un concept, un principe, une loi
- imaginer des dispositifs expérimentaux simples et prendre des initiatives
- Proposer une solution à l'énigme et la confronter avec la situation de départ
- Proposer au moins une piste de résolution possible
- Réfléchir aux pratiques mises en œuvre, évaluer la démarche suivie
- Confirmer ou infirmer un raisonnement par des arguments vérifiés
- Réaliser une brève communication orale, un petit exposé scientifique sur les résultats d'une recherche, en utilisant un média

Objectifs

L'élève sera capable de :

- Manipuler un système tangible (micro:bit)
- Utiliser une interface de programmation par blocs
- Comprendre le concept de variable : Comprendre le fonctionnement d'un ordinateur et principalement le stockage des données par ce dernier
- Comprendre le concept de structure conditionnelle : Comprendre le fonctionnement d'un ordinateur
- Trouver et reconnaître, dans l'interface de programmation, les blocs liés aux variables et structures conditionnelles





- Manipuler des variables (initialiser, modifier, afficher) dans un exercice donné
 - Manipuler des structures conditionnelles dans un exercice donné
 - Résoudre un problème donné et proposer une solution
 - Tester une solution
 - Comprendre pourquoi une solution ne fonctionne pas et la corriger
 - Reconnaître, à travers les solutions des autres, les variantes de solution possibles
 - Savoir expliquer sa solution (ses choix)
-

Description de l'activité

Un programme traite en général énormément d'informations. De ce fait, il est nécessaire de stocker ces informations quelque part et plus précisément, dans la mémoire, sous forme de variables. Il est également souvent utile de pouvoir dire *quand* un programme doit effectuer une certaine action. Les instructions conditionnelles permettent de le faire en indiquant une condition dans laquelle une action sera faite et éventuellement ce que le programme doit faire si la condition est fausse. En programmation, les variables et les conditions sont en quelque sorte les briques de bases sur lesquelles reposent tous les langages de programmation. Il est très difficile de faire un programme sans utiliser de variables ni de conditions.

Les élèves vont créer étape par étape un jeu du "plus ou moins" ce qui leur permettra de manipuler les deux nouveaux concepts.


Matériel et ressources

Matériel nécessaire :	Ressources nécessaires :
<ul style="list-style-type: none">• Par élève (ou groupe de 2)<ul style="list-style-type: none">– 1 micro:bit – 1 câble – 1 ordinateur• 1 ordinateur et un projecteur pour le professeur (ou tableau interactif)	<ul style="list-style-type: none">• fiche explicative "Variables"• fiche explicative "Conditionnelles"



Déroulement

Étape 1 : Les variables (10')


 *Si plus de temps est disponible, il peut être très intéressant de prévoir des activités en "débranché" pour découvrir les variables.*

Qui	Fait quoi	Matériel
Le professeur	Explique ce qu'est une variable	
Le professeur	Montre un exemple d'initialisation, de modification et d'utilisation de variable avec les blocs	ordinateur et projecteur

Étape 2 : Programmer le "plus ou moins" - 1ère partie (10')


Qui	Fait quoi	Matériel
Le professeur	énonce les fonctionnalités du jeu "plus ou moins"	
Le professeur	Distribue les fiches élèves où se trouvent l'énoncé et la marche à suivre	Fiches élève
Les élèves	Initialisent les variables au démarrage	ordinateur
Les élèves	Incrémentent en appuyant sur B et décrémentent en appuyant sur A	ordinateur
Les élèves	Testent leur solution et la corrigent si besoin	ordinateur (et micro:bit et câble USB)

Étape 3 : Les conditions (10')

 *Si plus de temps est disponible, il peut être très intéressant de prévoir des activités en "débranché" pour découvrir les conditionnelles.*

Qui	Fait quoi	Matériel
Le professeur	Explique ce que sont les conditions	
Le professeur	Montre un exemple de condition en décrémentant avec A seulement si le nombre est plus grand que 0	ordinateur et projecteur
Les élèves	Implémentent cette condition	ordinateur

Étape 4 : Programmer le "plus ou moins" - 2nde partie (10')

 *Si le temps manque, vous pouvez faire cette partie en réfléchissant tous ensemble.*

Qui	Fait quoi	Matériel
Les élèves	Implémentent que lorsqu'on secoue le micro:bit, il indique si c'est plus, moins ou si c'est gagné	ordinateur
Les élèves	Testent leur solution et la corrigent si besoin	ordinateur (et micro:bit et câble USB)

Étape 5 : Conclusion (5 min)

Qui	Fait quoi	Matériel
Tous	Discutent des différentes solutions possibles	


Pour aller plus loin

Les élèves peuvent apporter quelques modifications : modifier la borne (de 0 à 100) ou ajouter un nombre d'essai maximum autorisé (avec des boucles), enjoliver l'interface avec quelques animations

Solutions & Problèmes

Voici une solution pour l'initialisation des variables au démarrage :

```
au démarrage
définir mon_nombre à 0
définir random à choisir au hasard de 0 à 10
montrer nombre mon_nombre
```

 *"mon_nombre" peut avoir n'importe quel nom tant qu'on utilise le même pour la suite. Cependant, il faut essayer que le nom représente bien son utilité et éviter les accents. Pareil pour "random" qui signifie "au hasard" en anglais.*

Voici deux solutions possibles pour la décrémentation et l'incrémement. Elles font exactement la même chose toutes les deux. La première utilise le bloc "changer variable pour nombre" dont la traduction française est assez mauvaise et peut prêter à confusion pour les élèves. Il signifie plutôt "ajouter nombre à variable". La seconde solution sera peut-être plus naturelle pour certains.

```
lorsque le bouton A est pressé
changer mon_nombre pour -1
montrer nombre mon_nombre

lorsque le bouton B est pressé
changer mon_nombre pour 1
montrer nombre mon_nombre
```

Figure 1: Solution avec le bloc "changer variable pour..."

```
lorsque le bouton A est pressé
définir mon_nombre à mon_nombre - 1
montrer nombre mon_nombre

lorsque le bouton B est pressé
définir mon_nombre à mon_nombre + 1
montrer nombre mon_nombre
```

Figure 2: Solution en redéfinissant la variable

En repartant du code de la Figure 1, et en ajoutant une condition pour que mon_nombre ne soit jamais plus petit que 0 ni plus grand que 10, voici à quoi ressemble le code :

```
lorsque le bouton A est pressé
si (mon_nombre > 0)
alors changer mon_nombre pour -1
montrer nombre mon_nombre

lorsque le bouton B est pressé
si (mon_nombre < 10)
alors changer mon_nombre pour 1
montrer nombre mon_nombre
```

Cette solution fonctionne mais est moins "propre" (car il y a un trou) :



```
lorsque le bouton A est pressé
  si (mon_nombre <= 0)
  alors
  sinon
    changer mon_nombre pour -1
  montrer nombre mon_nombre
```

Pour la partie "secouer", voici deux solutions exactement équivalentes. La seule différence est que dans le second cas, on a utilisé le petit engrenage du "si" pour faire le "sinon si" en un seul bloc.

```
lorsque secouer
  si (mon_nombre < random)
  alors
    afficher texte "+ "
  sinon
    si (mon_nombre > random)
    alors
      afficher texte "- "
    sinon
      afficher texte "Bravo "

lorsque secouer
  si (mon_nombre < random)
  alors
    afficher texte "+ "
  sinon si (mon_nombre > random)
  alors
    afficher texte "- "
  sinon
    afficher texte "Bravo "
```



Certains élèves vont peut-être rajouter un "si" dans le dernier "sinon" pour vérifier que "mon_nombre" est égal à "random". C'est correct mais pas optimal : si c'est ni plus petit ni plus grand, c'est forcément égal ! Dès lors, le programme fonctionnera tout autant mais il devra tester une condition supplémentaire.

Pourquoi une solution pourrait ne pas fonctionner?

L'erreur la plus fréquente avec les conditionnelles est l'inversion des conditions. Faites également attention à utiliser correctement les "sinon, si".



Évaluation

Manipuler un système tangible (micro:bit) [./1]

	non	oui
Appuyer sur les boutons et secouer le micro:bit	0	1

Utiliser une interface de programmation par blocs [./0.5]

	non	oui
Ajouter et emboîter des blocs	0	0.5

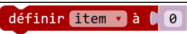
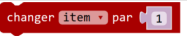
Comprendre le concept de variable : Comprendre le fonctionnement d'un ordinateur et principalement le stockage des données par ce dernier [./2]

	non	oui
Prendre conscience qu'une variable sert à stocker une valeur dans la mémoire	0	0.5
Prendre conscience que cette valeur peut être modifiée	0	0.5
Prendre conscience que cette valeur peut être récupérée	0	0.5
Prendre conscience que cette valeur peut être un nombre, du texte ...	0	0.5

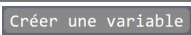
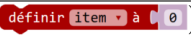
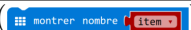
Comprendre le concept de structure conditionnelle : Comprendre le fonctionnement d'un ordinateur [./2]

	non	oui
Prendre conscience qu'une structure conditionnelle sert à faire une action si une condition est remplie (si... alors... (sinon...))	0	1
Prendre conscience que les conditions sont soit vraies soit fausses (aucune autre possibilité)	0	0.5
Prendre conscience que plusieurs structures conditionnelles peuvent se suivre (et être emboîtée)	0	0.5

Trouver et reconnaître, dans l'interface de programmation, les blocs liés aux variables et aux structures conditionnelles [./3]

	non	oui
Trouver le menu contenant les blocs liés aux variables	0	0.5
Trouver le menu contenant les blocs liés aux structures conditionnelles	0	0.5
Comprendre le bloc  (mettre une valeur dans une variable)	0	0.5
Comprendre le bloc  (ajouter une valeur à une variable)	0	0.5
Reconnaître les blocs de condition (si-alors et si-alors-sinon)	0	0.5
Savoir que les blocs '<' et '=' servent à comparer des éléments	0	0.5

Manipuler des variables (initialiser, modifier, afficher) dans un exercice donné [./2.5]

	non	oui
Créer une variable et l'initialiser ( et ()	0	0.5
Donner des noms significatifs aux variables	0	0.5
Ajouter 1 à une variable	0	0.5
Soustraire 1 à une variable	0	0.5
Afficher une variable ()	0	0.5



Manipuler des structures conditionnelles dans un exercice donné [./1.5]

	non	oui
Utiliser un si-alors-sinon	0	0.5
Utiliser une comparaison comme condition	0	0.5
Emboîter un 'si' dans un 'sinon'	0	0.5

Résoudre un problème donné et proposer une solution [./3.5]

	non	oui
Créer 2 variables et les initialiser	0	0.5
Initialiser une variable avec un nombre aléatoire	0	0.5
Ajouter 1 à une variable lorsque le bouton B est pressé	0	0.5
Soustraire 1 à une variable lorsque le bouton A est pressé	0	0.5
Si le nombre est plus petit que celui à trouver, afficher "+"	0	0.5
Si le nombre est plus grand que celui à trouver, afficher "-"	0	0.5
Programmer le jeu complet	0	0.5

Tester une solution [./1]

	non	oui
Tester la solution (sur le micro:bit ou l'émulateur)	0	1

Comprendre pourquoi une solution ne fonctionne pas et la corriger [./1]

	non	oui
Ne pas avoir d'erreur ou les corriger s'il y en a	0	1

Reconnaître, à travers les solutions des autres, les variantes de solution possibles [./1]

	non	oui
Voir les différences avec d'autres codes qui fonctionnent	0	1

Savoir expliquer sa solution (ses choix) [./1]

	non	oui
Expliquer sa solution aux autres	0	1