

Fiche d'activité Botly : Cycle Initiation

Séance 3 : Les variables

Durée : environ 40 minutes

Résumé de l'activité : Intégrer la notion de variable et son intérêt dans un contexte de programmation

Objectifs Pédagogiques :

- Appréhender les notions de base de la programmation (instruction, paramètre, succession d'instruction)
- Aborder des notions géométriques (angle, forme,...) par le jeu et les proportions
- Maîtriser la notion de variable

Participants : Entre 8 et 16 participants (2 élèves par ordinateur)

Matériel : 4 à 8 ordinateurs avec Botly-Studio, ou disposant d'une connexion internet, si possible un robot pour la présentation

Préparation : Disposer 8 ordinateurs disposant d'une connexion internet et ouvrir botly-studio.fr dans un navigateur (**MODE : DEBUTANT**)

Notions liées : Instruction, Bloc, Programme, Simulation, Botly, Boucle, **Variable**

Introduction et rappel : 5'

Bilan rapide de la séance précédente sur les boucles.

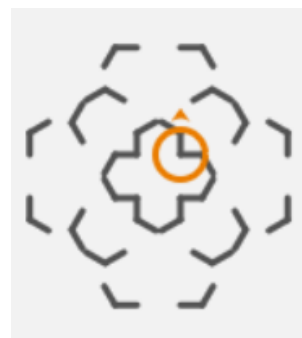
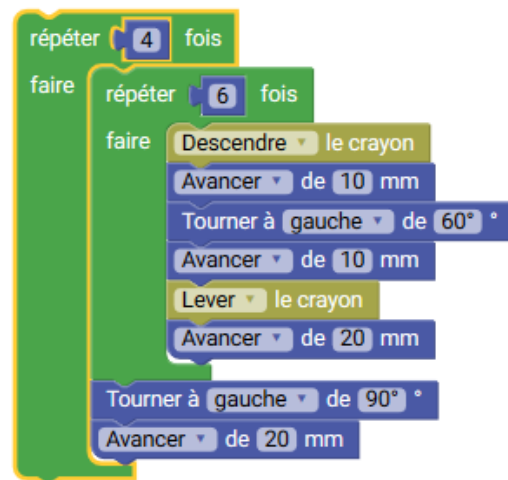
Le système des boucles est utilisé pour gagner du temps en programmation lorsqu'il est nécessaire de faire **plusieurs fois la même opération**.

On peut faire varier **le contenu** d'une boucle, mais également **le nombre de répétition**.

On peut **imbriquer une boucle dans une autre** pour reproduire des formes ou des motifs par exemple.

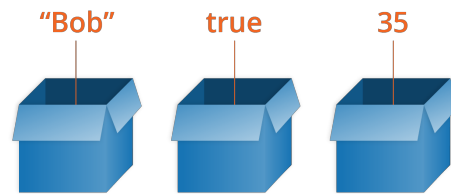
Si les élèves ont utilisé le logiciel entre 2 séances pour s'exercer, on peut demander de montrer ses créations, notamment celles utilisant les **boucles imbriquées**.

Exemple : *Mosaïque*

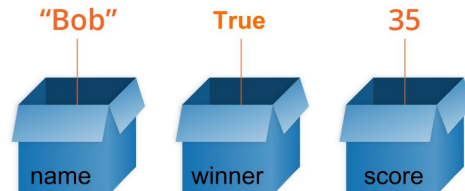


Introduction de la notion de variable - 15'

En programmation, une variable est **comme une boîte** dans laquelle on peut stocker une information.



Au début d'un programme, on va créer une boîte en lui donner **un nom**, et **y placer quelque chose**, comme un chiffre par exemple. C'est ce que l'on appelle **déclarer une variable**.

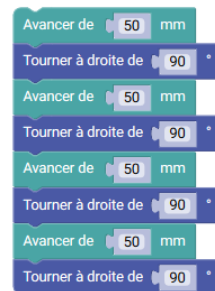


Par la suite, il suffira d'utiliser le nom de sa boîte pour accéder à son contenu. Cela peut être utile, notamment lorsque l'on a besoin d'utiliser plusieurs fois la même information à différents endroits.

Prenons un exemple simple. Pour dessiner un carré de 50 mm de côté, je peux utiliser les blocs classiques :



Je dois alors modifier les blocs "Avancer de XX mm" un à un pour obtenir 4 côtés de 50 mm. Si je veux ensuite en réaliser un de 40 mm, je dois à nouveau modifier les blocs "Avancer de XX mm" un à un.



Pour 50 mm



Pour 40 mm

Expérimenter via une suite de challenges - 20'

La notion n'est pas simple à appréhender et l'intérêt de l'utiliser pour des formes de base n'est pas évident. Essayons donc d'illustrer l'utilité de la variable par une suite de défis.

Défi 1 : Dessiner une maison à taille variable

Pour ce premier défi, l'objectif est de construire une maison simplifiée dont on peut facilement modifier la taille en 1 opération.

En ce qui concerne la forme, la maison est constitué de 2 formes : un carré et un triangle. Pour que cela fonctionne, il faut que les 3 formes fassent la même taille. Il faut donc ré-utiliser les codes pour le triangle et pour le carré.

Mais il faut également que l'on puisse modifier facilement la taille de ces 2 formes par la même opération. Il va donc falloir utiliser une variable "longueur" qui soit commune aux 2 formes.



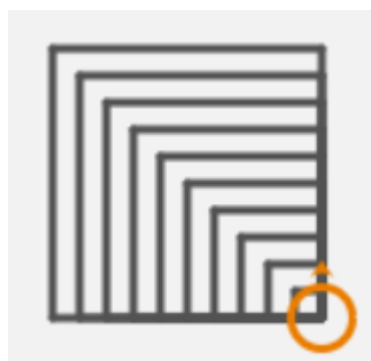
Corrigé :

fixer longueur à 70	Déclaration de la variable "longueur"
Tourner à droite de 90 °	Rotation pour mettre la maison dans le bon
répéter 4 fois faire Avancer de longueur mm Tourner à droite de 90 °	Code correspondant au carré formant la base de la maison. La taille des côtés est définie par la variable "longueur".
répéter 3 fois faire Avancer de longueur mm Tourner à gauche de 120 °	Code correspondant au triangle formant le toit de la maison. La taille des côtés est également définie par la variable "longueur".

Ici, l'intérêt de la variable est de pouvoir contrôler facilement la taille de sa maison en 1 opération. Il suffit de modifier le contenu de la "boîte" (la variable "longueur") pour obtenir une maison plus grande ou plus petite.

Défi 2 : Faire une suite de forme de taille croissante

Dans le deuxième défi, l'objectif est de construire un code permettant de tracer 10 carrés à la suite dont la taille augmente de 10 mm passant de 10 mm à 100 mm. Concrètement, la forme que l'on cherche à atteindre ressemble à celle-ci :

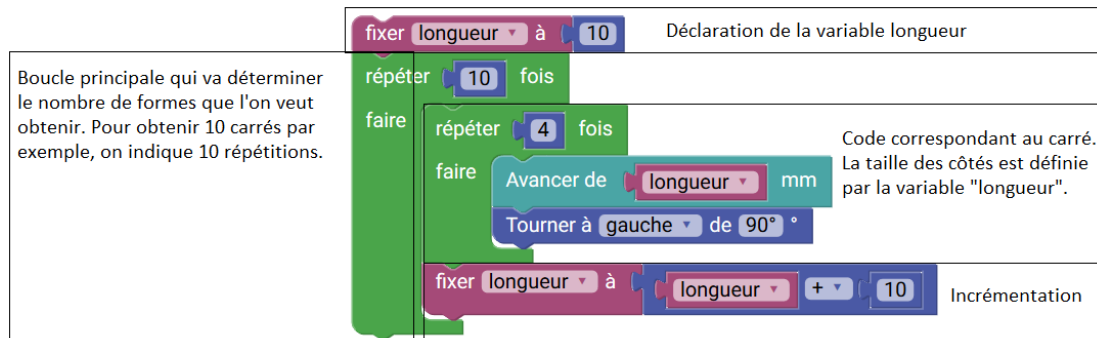


Pour réaliser ce code, il faut décomposer la consigne de l'exercice. Il faut tracer un carré, qui a **4 côtés identiques**.

Il faut ensuite réaliser 10 carrés, donc **répéter 10 fois la même figure**.

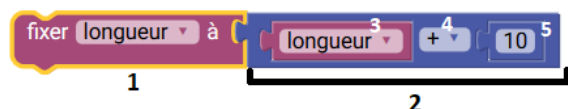
Enfin, il faut faire en sorte que la taille des carrés augmente progressivement à chaque fois que l'on en trace un nouveau.

Pour cela, on peut utiliser ce que l'on appelle en programmation l'**incrément** (c'est-à-dire l'augmentation de la valeur (d'une variable) à chaque exécution d'un programme).



1. Comme dans le défi précédent, on commence par définir une variable que l'on nomme "longueur". On place la valeur 10 dans la variable puisque notre premier carré doit avoir une longueur de 10 mm.
2. Ensuite, à la différence du défi précédent, il faut réaliser 10 fois la même forme. On peut donc utiliser une boucle pour faire répéter 10 fois les mêmes instructions.
3. Puisque les formes que l'on souhaite obtenir sont des carrés, on utilise le code correspondant au carré qui a été vu précédemment. Comme dans le défi précédent, on utilise la variable "longueur" pour contrôler la taille du carré.
4. La principale difficulté de cet exercice est l'incrément. Il s'agit d'une opération qui consiste à modifier le contenu de la variable "longueur" à chaque fois qu'une boucle est parcourue par le programme.

Concrètement, voici l'instruction qui correspond à cet outil :



Puisque l'on souhaite modifier le contenu de la variable "longueur", on va utiliser le bloc "fixer "longueur" à (1).

Ce que l'on souhaite ce n'est pas lui attribuer une nouvelle valeur (par exemple 20), mais ajouter 10 à la valeur déjà contenue dans la variable, on utilise donc un bloc permettant d'additionner (2).

Il ne reste ensuite qu'à écrire l'opération : on sélectionne la variable "longueur" (on "ouvre la boîte") (3), on ajoute le signe "+" (4) et on ajoute le nombre "10" (5).

Ainsi à chaque fois que le programme va arriver à cette ligne, il va ouvrir la variable longueur, et ajouter 10 à la valeur déjà contenue dedans.

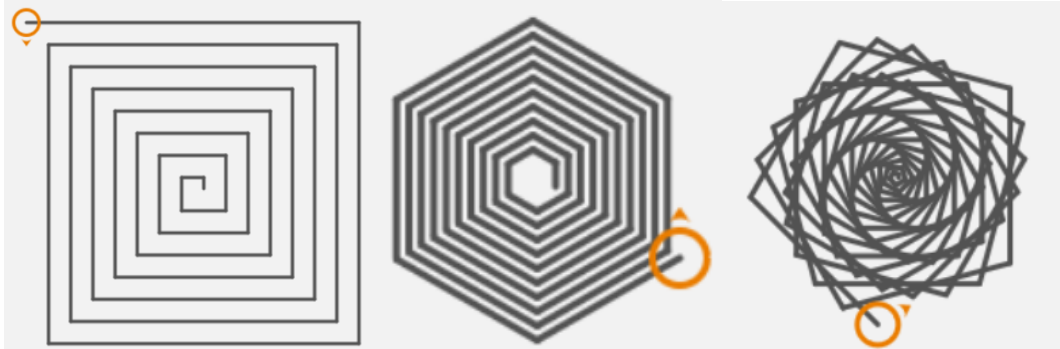
Au premier passage, on aura donc :

- "longueur" = 10
- + 10
- "longueur" = 20

Au deuxième passage, on aura :

- "longueur" = 20
 - + 10
 - - "longueur" = 30
- etc.

Ce système d'incrémentation est très utilisé en programmation. Pour le dessin, il permet par exemple de réaliser des spirales .



Activité : pour finir la séance, il est possible de laisser les participants explorer les différents types de spirales possibles. L'exercice est assez fascinant et donne envie d'explorer les différents combinaisons possibles.

Conclusion

Bilan

Qu'est-ce que j'ai appris ? Qu'est-ce qui était difficile ?

A retenir

- En programmation, une variable est **comme une boîte** dans laquelle on peut stocker une information.
- Au début d'un programme, on va créer une boîte en lui donner **un nom**, et **y placer quelque chose**, comme un chiffre par exemple. C'est ce que l'on appelle **déclarer une variable**.
- Un variable peut être utile pour **contrôler la taille de plusieurs objets** en une opération.
- Elle est utilisée pour la technique de **l'incrémentation** pour tracer des spirales par exemple.

Trace écrite

A prévoir : un document qui résume les concepts, et montre le code commenté correspondant ainsi que la figure dessinée