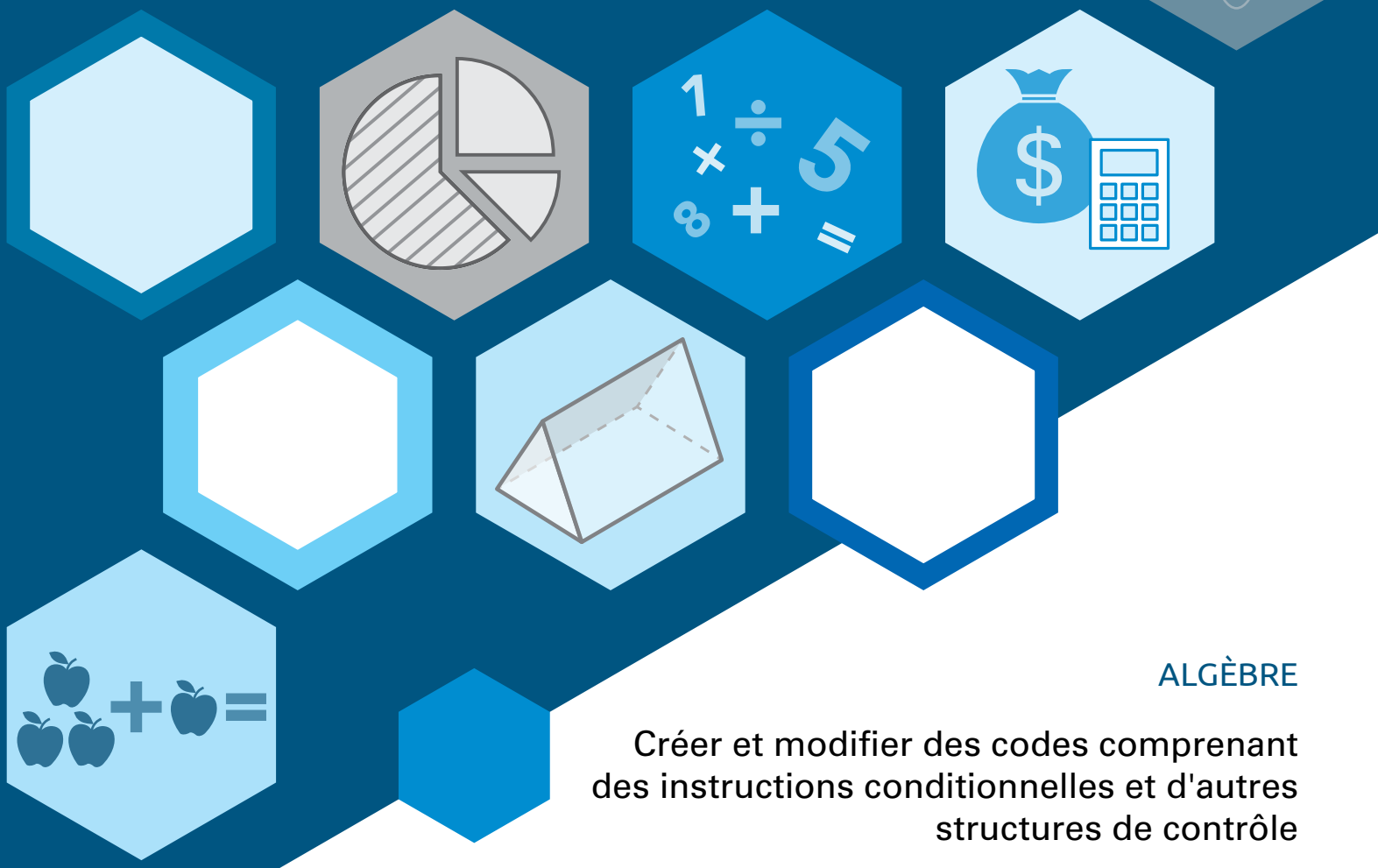
5e année

En avant, les maths!

Une approche renouvelée pour l'enseignement et l'apprentissage des mathématiques

MINILEÇON



ALGÈBRE

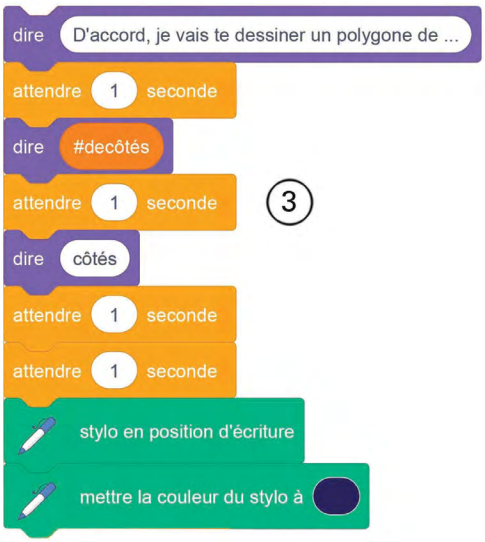
Créer et modifier des codes comprenant des instructions conditionnelles et d'autres structures de contrôle

EXEMPLE 1

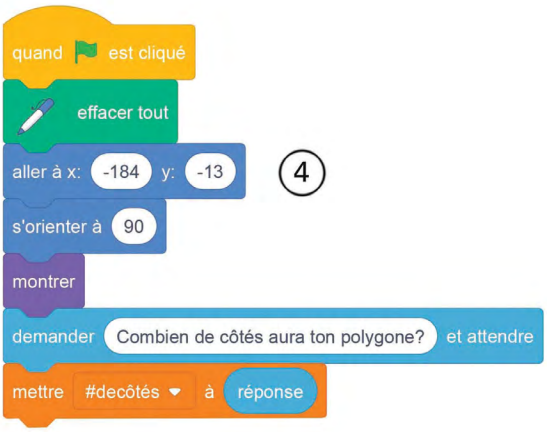
Ces codes viennent d'un programme *Scratch* qui demande à l'utilisateur de choisir le nombre de côtés d'un polygone et qui dessine un polygone. Peux-tu remettre les codes dans le bon ordre afin de créer un code qui contient une instruction conditionnelle?

1

2

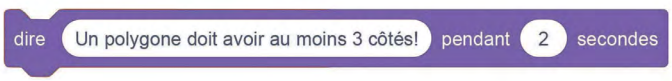


3



4

5



On obtient l'extension du stylo en cliquant sur son icône en bas à gauche de l'écran.

STRATÉGIE

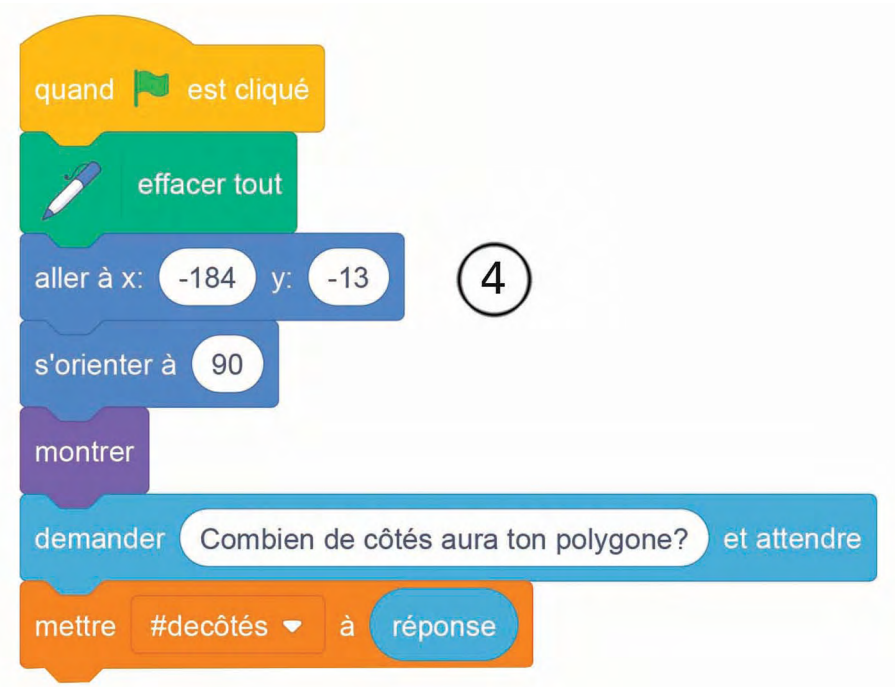
Recréation du programme dans le bon ordre

Voici un cheminement possible afin de résoudre le problème et ainsi recréer le programme dans le bon ordre.

Je remarque qu'il y a 5 différentes parties identifiées par les chiffres 1, 2, 3, 4 et 5.

Je les lis pour identifier ce que chacune des parties faisait dans le programme original.

La partie 4 du programme :

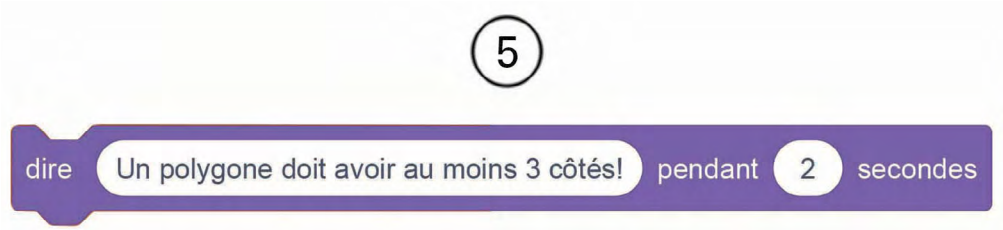


- contient un bloc « démarrer » qui est unique, alors doit aller au début;
- pose une question à l'utilisateur et lui demande de donner une réponse.

La partie 2 du programme :

- contient l'instruction conditionnelle. Il y a 2 options, mais il n'y a pas de blocs dans les 2 parties après « si » et « sinon ». Le premier « si » est défini pour toutes les réponses qui sont inférieures à 3.

La partie 5 du programme :



- correspond à la réponse qui serait plus basse que 3, par exemple 1 ou 2, puisqu'un polygone doit avoir 3 côtés ou plus;
- je dois donc placer ce bloc dans l'espace blanc tout de suite après « Si réponse est inférieure à 3 alors ».

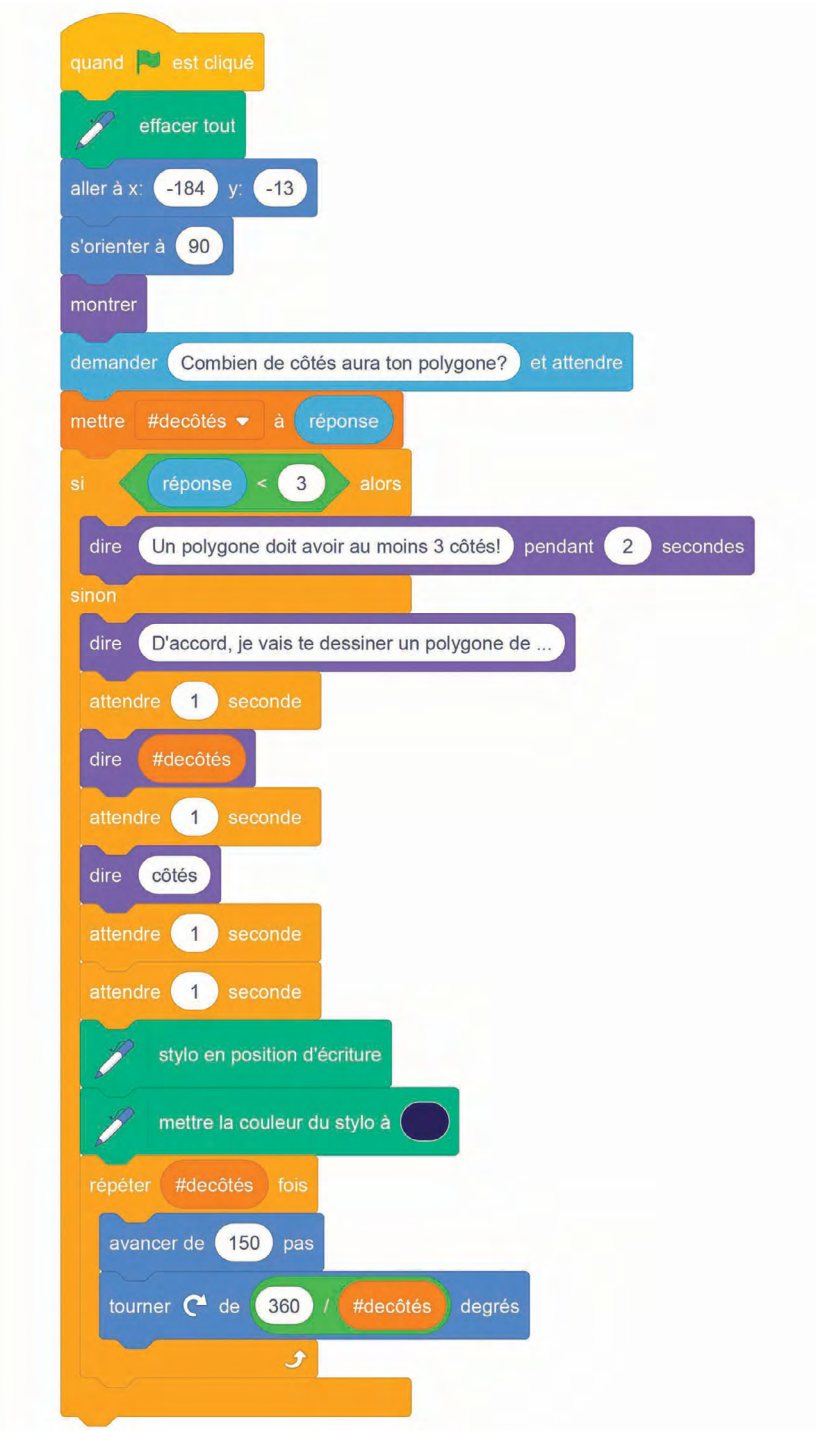
La partie 3 du programme :

- « dit » que le programme va dessiner un polygone et de combien de côtés en réponse à la question posée au début du programme;
- prépare le stylo en le déposant sur le papier et en ajustant sa couleur (on obtient l'extension du stylo en cliquant sur son icône en bas à gauche de l'écran);
- doit aller dans la deuxième partie de l'instruction conditionnelle après « sinon »;
- le programme est donc prêt à dessiner le polygone!

La partie 1 du programme :

- est la partie qui dessine le polygone;
- elle doit aller à l'intérieur de l'instruction conditionnelle immédiatement après le numéro 3 puisque le stylo doit avancer pour dessiner le polygone choisi par l'utilisateur.

Voici le programme complet que j'ai recréé :



EXEMPLE 2

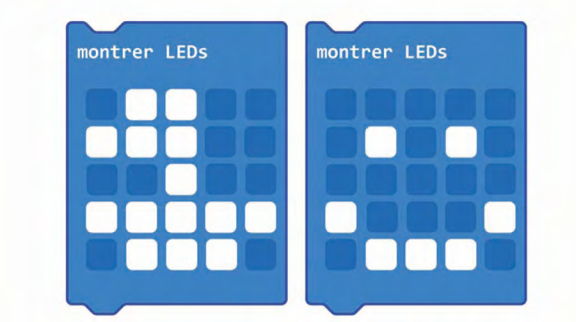
As-tu déjà utilisé une pièce de monnaie pour faire « pile ou face » en attrapant la pièce et en la posant sur le dos de ta main?

Un logiciel en ligne tel que makecode.microbit.org peut te permettre de simuler la même activité pour choisir entre deux options.

Crée un programme simple « Pile ou face » en 5 étapes :

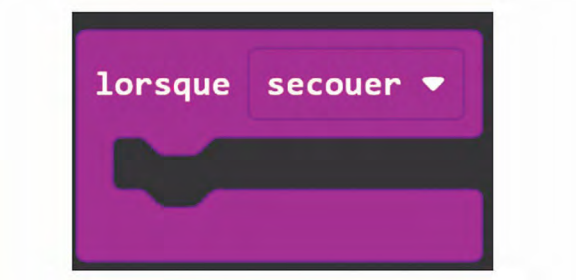
Étape 1 :

Utilise les blocs « Base » pour créer une image « pile » et une image « face ». Voici 2 exemples, mais tu peux créer ton propre dessin « pile » ou « face ». Ces dessins correspondent aux images sur la pièce de 10 ¢ canadienne.



Étape 2 :

Choisis une entrée. Tu peux décider d'utiliser un bouton, ou une entrée « secouer » dans les boutons « entrée ». L'entrée est le bouton ou l'action qui va démarrer le programme.

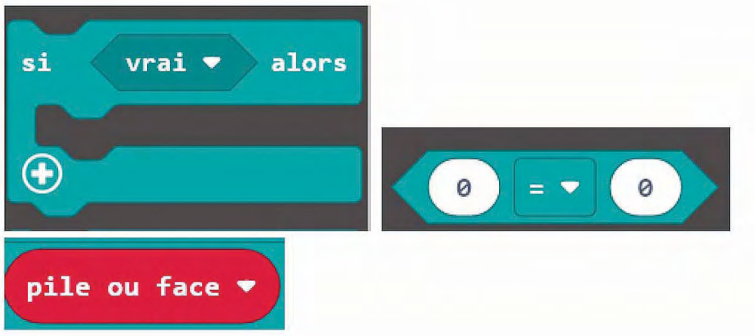


Étape 3 :

Tu dois maintenant créer une variable « pile ou face » qui va être égale à 1 ou 2. Dans ton programme, « 1 » va être pile et « 2 » va être face. Tout ceci peut être fait sous les blocs « variables » et « maths ». Voici la ligne de blocs que tu dois trouver, créer et assembler. Attention à la forme et à la couleur des blocs!

Étape 4 :

Crée l'instruction conditionnelle en combinant les 3 blocs illustrés ci-dessous. Celle-ci se trouve dans les blocs turquoise intitulés « logique ». Tu dois insérer le bloc d'égalité pour avoir les options 1 ou 2 (puisqu'il n'y a qu'une option autre que « 1 », 2 n'a pas besoin d'être nommé dans le programme). Le bloc « pile ou face » est celui que tu avais créé à l'étape 3.



Étape 5 :

Assemble tous tes blocs et teste ton code!

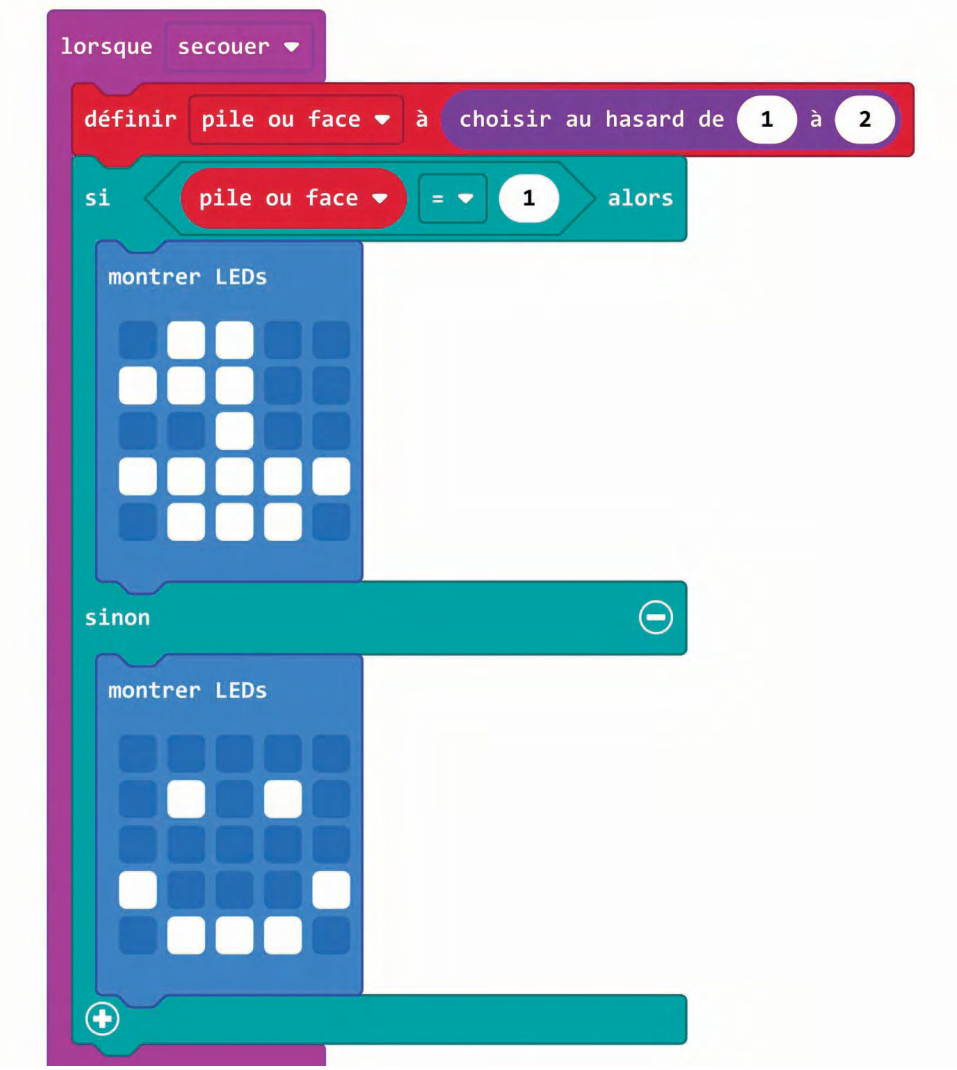
STRATÉGIE

Identification des couleurs du code et respect des étapes à suivre

J'identifie les couleurs du code dans les étapes et je suis les étapes une à une en séquence afin de ne rien oublier :

1. Je dessine les blocs de la première étape et je les mets de côté pour plus tard. Je les ai trouvés dans la catégorie bleue « Base ».
2. Je choisis mon entrée dans la catégorie de blocs rose « Entrée ».
3. Je reproduis avec précision le code de l'étape 3 dans la catégorie rouge « Variables ».
4. J'assemble les formes des 3 blocs de l'étape 4 pour qu'elles s'emboîtent les unes dans les autres.
5. Pour l'assemblage, je décris ce qui va arriver dans mon programme dans mes propres mots pour m'assurer que les blocs sont dans le bon ordre :
   - Lorsque je secoue le micro:bit, il va choisir une des deux options possibles que j'ai nommées 1 ou 2 dans la variable « pile ou face ».
   - Si la variable aléatoire est 1, alors le programme va montrer mon petit bateau (comme la pièce de 10 ¢).
   - Si la variable aléatoire est autre que 1 (2), alors le programme va montrer le bonhomme sourire.

Voici le programme que j'ai créé en suivant les étapes :

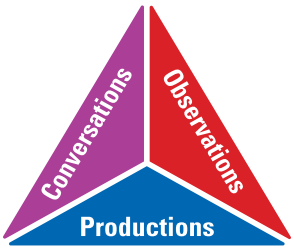


PARTIE 2 – PRATIQUE AUTONOME

Déroulement

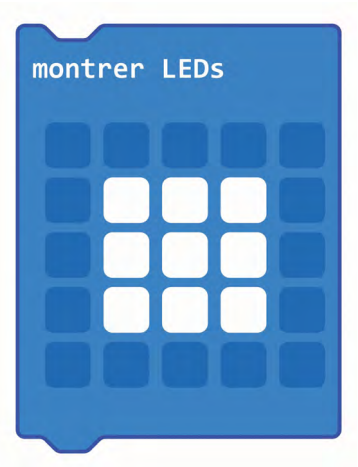
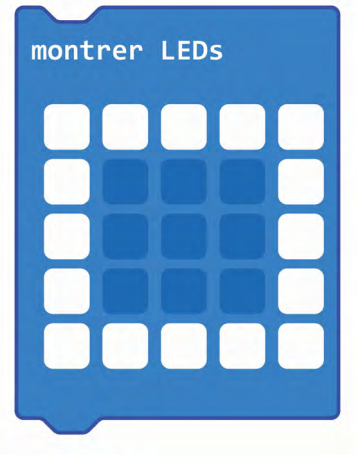
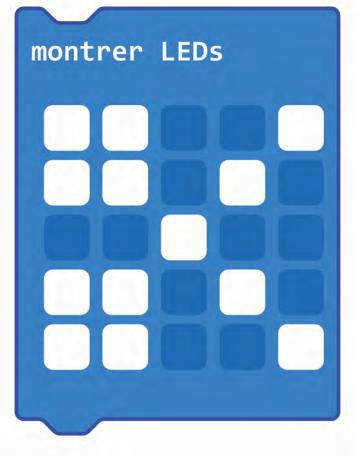
- Au besoin, demander aux élèves de faire quelques exercices de la section **À ton tour!**. Ces exercices peuvent servir de billet de sortie ou autre.
- Recueillir les preuves d'apprentissage des élèves et les interpréter pour déterminer leurs points forts et cibler les prochaines étapes en vue de les aider à s'améliorer. Afin de recueillir des preuves d'apprentissage tout au long du processus de codage, poser des questions telles que :
  - Comment as-tu su que tu y étais arrivé?
  - As-tu eu des difficultés avec certaines fonctions?
  - Où se trouve tes instructions conditionnelles?
- Demander aux élèves de partager leurs projets en envoyant le lien URL.
- Dans la plupart des programmes, les élèves peuvent écrire un commentaire associé aux blocs qu'ils utilisent et qui explique leur utilité dans le programme. Encourager les élèves à laisser des traces en utilisant cette fonctionnalité.

  Note : Consulter le corrigé de la partie 2, s'il y a lieu.

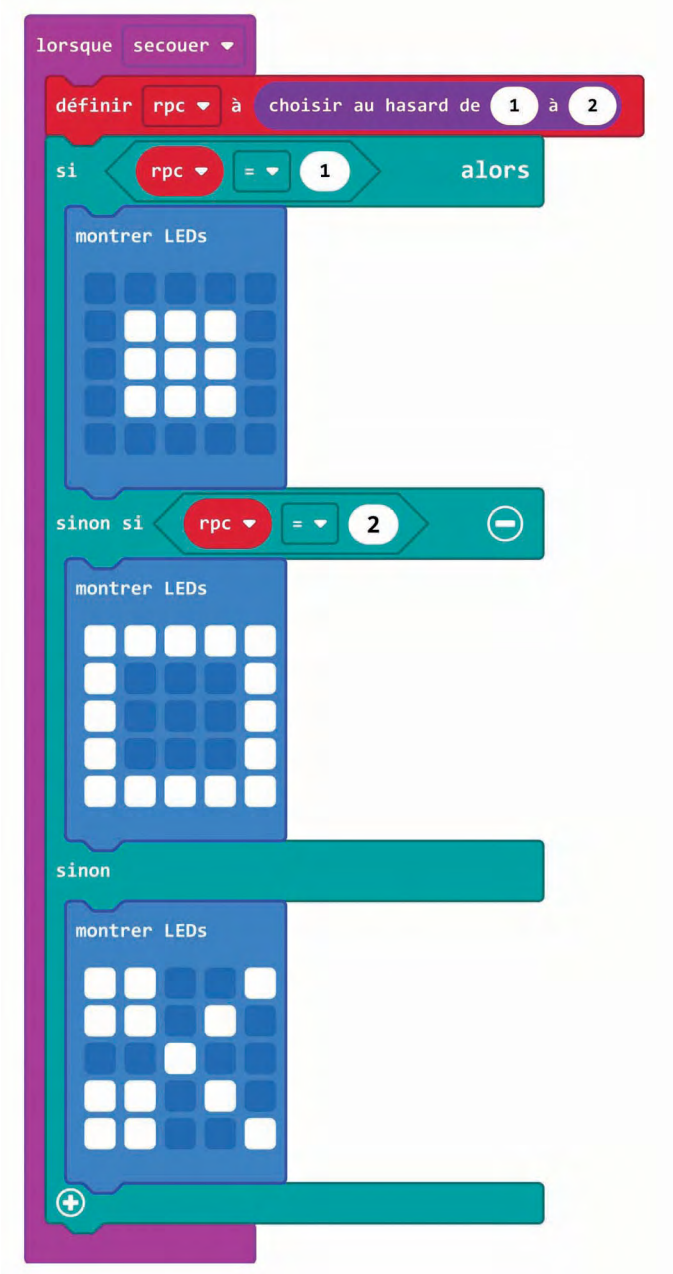


CORRIGÉ

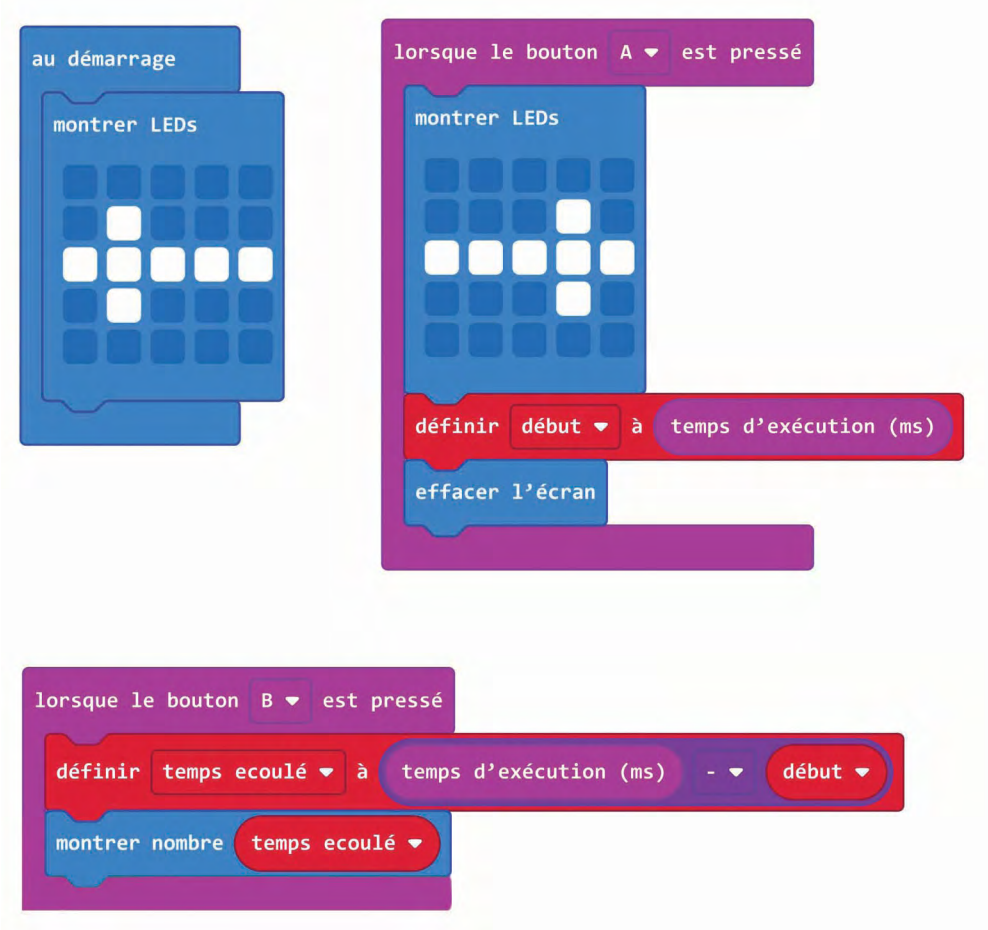
1. Maintenant que tu sais comment créer un code qui contient une instruction conditionnelle et d'autres structures de contrôle, tu peux t'exercer à créer de nouveaux codes avec des instructions conditionnelles plus complexes. Par exemple, en ajoutant une troisième option dans un code, on peut créer un jeu qui affiche roche, papier ou ciseaux pour jouer avec un ami qui a créé le même programme que nous! Utilise ce tableau pour créer ton code en utilisant un logiciel tel que makecode.microbit.org.

Objet	Variable roche-papier-ciseaux	Image à afficher	Position dans les instructions
Roche	1		Si
Papier	2		Sinon si… alors
Ciseaux	3		Sinon

Voici un exemple d'un programme qui permet de jouer à roche-papier-ciseaux avec un micro:bit ou dans le logiciel.



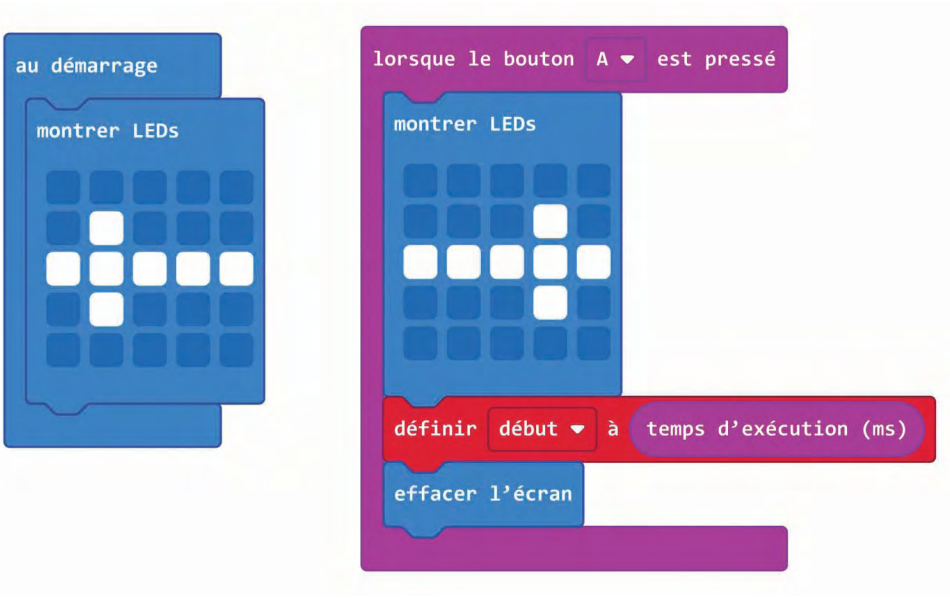
2. Voici un programme qui mesure ton temps de réaction en millisecondes. Afin de jouer, tu dois appuyer sur le bouton « A » pour démarrer une minuterie et appuyer sur le bouton « B » pour l'arrêter. Le programme affiche alors ton temps de réaction en millisecondes et affiche le résultat (programme d'action/réaction créé avec makecode.microbit.org) :

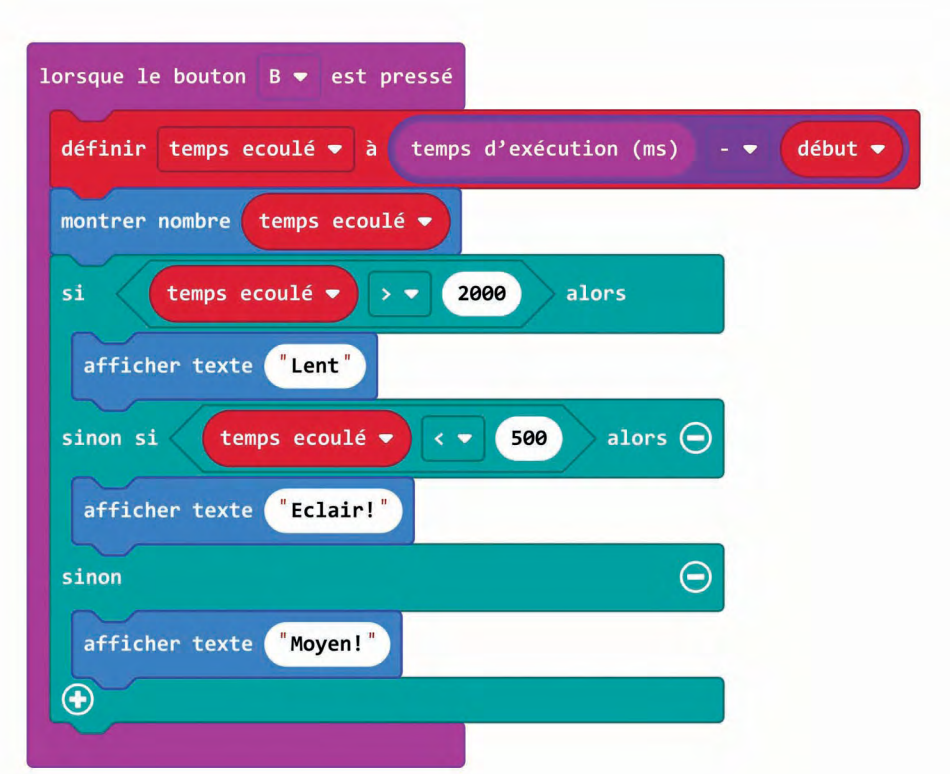


Peux-tu améliorer le programme en y ajoutant une instruction conditionnelle qui va mesurer le temps de réaction et donner un résultat basé sur le barème suivant à la personne qui joue?

Temps de réaction	Mot à afficher
2 secondes ou plus	Lent
De 500 ms à 2 secondes	Moyen
De 0 à 499 millisecondes	Éclair!

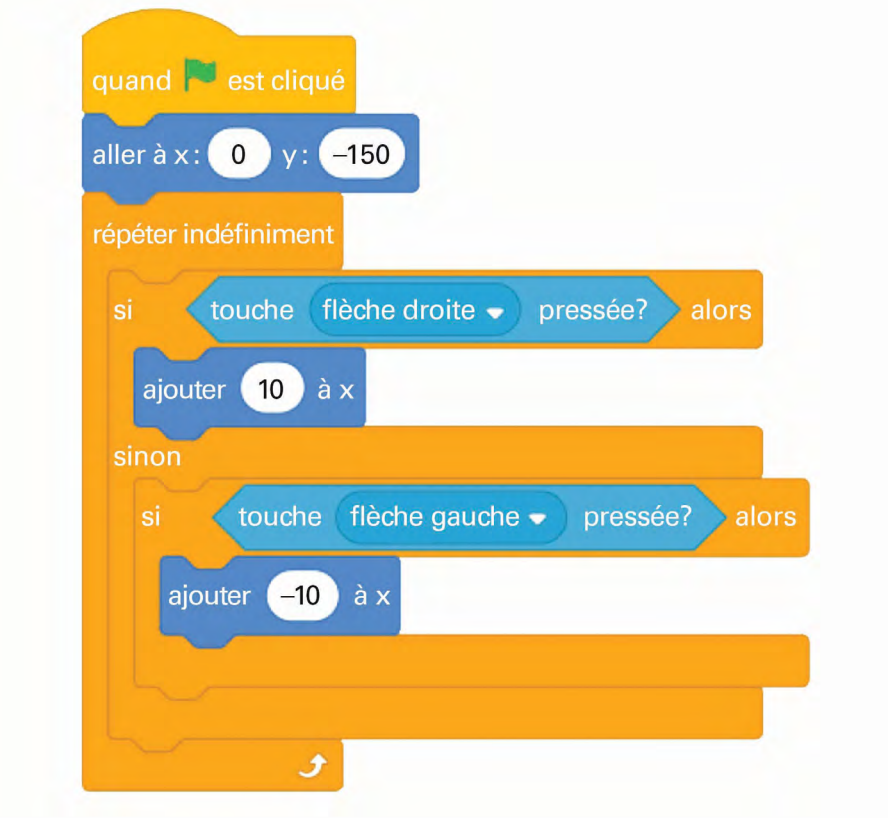
Voici un exemple d'un code qui permet d'ajouter une instruction conditionnelle pour évaluer la rapidité du temps de réaction du programme :





3. Crée un programme de ton choix qui incorpore une instruction conditionnelle.

   Voici un exemple : J'ai créé un programme avec *Scratch* qui permet de faire bouger le chat de gauche à droite dans le programme :



4. Dans le programme que tu viens de créer, quelle est l'instruction conditionnelle?

   L'instruction conditionnelle est que si la personne qui utilise le programme appuie sur la flèche droite, le chat va bouger vers la droite de 10 pas, mais si la personne appuie sur la flèche gauche, le chat va bouger de 10 pas vers la gauche. Il y a aussi une condition implicite qui est que si la personne n'appuie pas du tout, le chat ne va pas bouger du tout!

5. Dans ton programme, est-ce qu'il y a une structure de contrôle autre que l'instruction conditionnelle? (Souviens-toi qu'une structure de contrôle, c'est une ligne ou un bloc de code qui influence l'ordre, par exemple une boucle ou une séquence).

   Oui, il y a une structure de contrôle puisque dans le programme, en plus de l'instruction conditionnelle, il y a l'instruction du début qui est une séquence et aussi une boucle infinie. Quand la personne veut commencer le programme, elle doit appuyer sur le drapeau pour commencer le programme. Ensuite, il y a une boucle infinie qui attend que l'on exécute les conditions dans le programme.