
Initiation à la programmation : coder avec Scratch

— guide de l'animateur —

Bienvenue à toi animateur Scratch !

Te voilà embarqué dans la grande aventure de la transmission du savoir-coder ! :)

Ce guide est là pour t'aider à réaliser ton atelier en toute sérénité. Tu y trouveras un déroulé type pour un atelier d'1 heure.

Qui dit “déroulé type” ne dit pas “déroulé fixé dans le marbre”. A toi de te l'approprier, de l'adapter en fonction du temps qui t'es imparti et de la tranche d'âge que tu vas former.

In code we trust ! ;)

Contextualisation (10-15 minutes)

Comment expliquer la programmation aux enfants ?

Introduire la logique de programmation

Apprendre à coder avec Scratch (30-35 minutes)

C'est quoi Scratch ?

Première chose à faire : s'inscrire sur Scratch !

Scratch, ça marche comment ?

Se repérer sur une page projet

Animer un personnage

Les boucles et les conditions avec Scratch

Personnalisation des projets (10-15 minutes)

CONTEXTUALISATION

Avant de mettre l'enfant sur l'ordinateur, il est important de replacer l'atelier dans son contexte. C'est quoi la programmation ? C'est quoi une boucle ? C'est quoi une condition ?

*Des questions qu'il est **essentiel** d'aborder avant même de commencer à coder pour le bon déroulé de l'atelier, surtout pour les primaires.*

*Si vous sautez cette étape, vous aurez beaucoup de mal à les faire programmer quoi que ce soit. Cette première partie d'atelier se déroule donc avec des écrans **éteints** (important si vous voulez avoir toute leur attention).*

*C'est une introduction, sa durée ne doit donc pas excéder **15 minutes pour un atelier d'une heure**. Adaptez le temps consacré à cette première partie selon la tranche d'âge.*

Comment expliquer la programmation aux enfants ?

Apprendre la programmation c'est bien, mais qu'est-ce que c'est ? A quoi ça sert ?

On retrouve la programmation dans les ordinateurs, les téléphones... mais on ne voit vraiment le code qui la compose que rarement. Comment ça marche tout ça ?

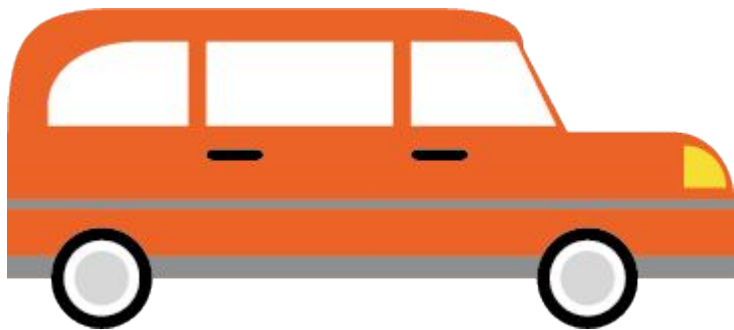
Comparons un site, une application ou (dans le cas présent) un jeu à une voiture :

Lorsqu'on regarde une voiture on ne voit qu'une machine qui roule. Même lorsque l'on soulève le capot pour observer le moteur, à moins de s'y connaître on ne voit que des câbles et des pistons.

Pourtant il y a un moteur qui fait tourner des roues et tout un système mécanique complexe.

C'est le mécanicien qui fait en sorte de faire fonctionner ce moteur et de réparer la voiture quand celle-ci est cassée.

Un développeur c'est un peu comme le mécanicien du code, celui qui saura comprendre la machinerie sous le capot :)



Voyons donc comment fonctionne le code et apprenons à devenir des apprentis mécaniciens du web en faisant un petit jeu ou une animation sur Scratch ! :)

Introduire la logique de programmation

Le code c'est de la logique. Si je fais telle action mon personnage devra faire tel mouvement dans telle direction.

Ce moment de l'atelier est important car vous allez aborder des concepts compliqués comme **les boucles** et **les conditions**. Et surtout vous allez devoir expliquer ce que sont **les abscisses** et **les ordonnées**.

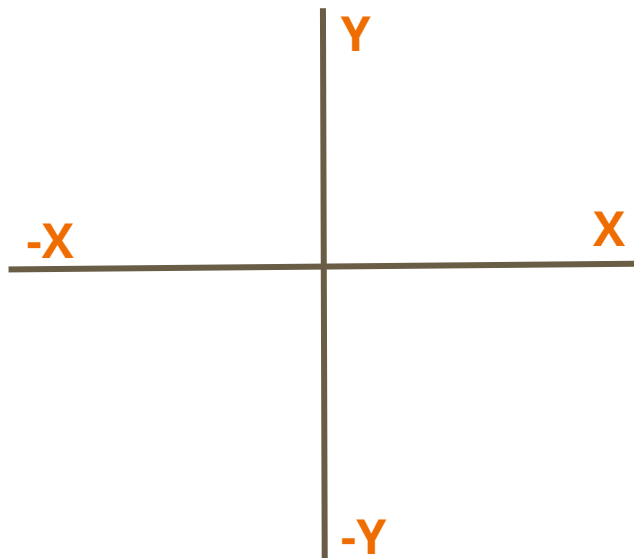
Essayons donc déjà en vrai avant de se lancer dans le code.

Demandez un volontaire pour faire une illustration IRL :)

Si vous sentez la classe un peu timide (ce qui est possible en début d'atelier, on ne se connaît pas, on ose pas aller devant les copains), n'insistez pas trop et n'hésitez pas à servir vous-mêmes de cobaye ;) Quitte à redemander un volontaire plus tard.

C'est quoi les abscisses et les ordonnées ?

Commençons par dessiner **une grande croix**. Ajoutons **X**, **-X**, **Y** et **-Y** comme ceci :



Demandez aux enfants de prendre une feuille, ou utilisez un tableau si vous en avez un à disposition.

Expliquez aux enfants que le **X** indique au volontaire d'aller à **gauche**, le **-X** à **droite**, le **Y** en **haut** et le **-Y** en **bas**.

Maintenant faisons un petit jeu :)

Le volontaire est un personnage de jeu qui va se faire diriger par ses camarades !

Les enfants doivent lui dire de bouger vers X (de faire un pas sur la droite), vers Y (de sauter), vers - Y (de se baisser) ou vers -X (de faire un pas sur la gauche).

Quand vous voyez que le jeu prend bien, n'hésitez pas à compliquer le tout en demandant de faire une combinaison d'actions (comme quand on joue aux jeux vidéos!) : X, X, -Y, Y.

C'est quoi une condition ? C'est quoi une boucle ?

Corsons un peu les choses. On sait comment faire se déplacer un personnage, mais on aimerait bien aller un peu plus loin en le faisant interagir avec d'autres éléments. Comment faire ?

Tout simplement en utilisant **une condition**.

Par exemple : **si** je tape des mains, **alors** mon personnage bouge 1 fois de - X.

Amusez-vous à trouver différentes conditions avec les enfants pour faire bouger le volontaire, afin d'être sûr qu'ils retiennent bien ce concept.

Utilisez bien les mots "si...alors..." dans ces exemples car ce sont les termes utilisés dans le script de la bibliothèque Scratch. Il sera ensuite plus facile pour vous de faire le lien.

Maintenant que j'interagis avec mon personnage, je voudrais bien qu'il répète son action plusieurs fois.

Pour cela nous allons utiliser **les boucles**.

Par exemple : **Répéter** 10 **fois** de bouger vers X.

Et qu'est-ce qui se passe si on fait répéter l'action à son personnage **sans lui dire quand s'arrêter** ?

Essayons ! Dîtes au volontaire : **Répéter indéfiniment** bouger vers Y.

Le personnage saute sans s'arrêter ! C'est ce qu'on appelle **une boucle infinie**.

Comme pour les conditions utilisez bien les mots "répéter X fois" et "répéter indéfiniment" dans ces exemples pour faire le lien avec les blocs Scratch qui leur permettront de faire des boucles.

Et ce qu'il y a de génial avec les boucles et les conditions, c'est qu'elles peuvent fonctionner ensemble !

L'épreuve ultime, demandons à notre personnage :

Si je dis "Hey" **alors**

Répéter 5 **fois** bouger vers -X

Vous êtes libre de choisir les exemples que vous voulez. N'hésitez pas à être créatif et à motiver les enfants, une fois ces concepts bien ancrés la suite de l'atelier n'en sera que plus simple pour vous :)

Maintenant qu'on a compris tout ça, on peut passer au code !

APPRENDRE À CODER AVEC SCRATCH

*Les notions abstraites les plus importantes ont été abordées lors de la première partie de l'atelier. Il ne reste plus qu'à se lancer et à voir comment animer un personnage sur Scratch ! Dans l'idéal, sur un atelier d'1h il faudrait que cette partie n'excède pas **30-35 minutes**. Cela dépendra beaucoup de la tranche d'âge avec laquelle vous allez faire l'atelier.*

N'hésitez pas à bien prendre le temps de leur montrer tous les outils qu'ils ont à disposition sur la plateforme pour qu'ils l'exploitent au maximum.

*Ayez des attentes réalistes sur les projets que les enfants vont faire : il est peu probable qu'un enfant de 8 ans ait le temps ou la capacité de coder un Call of Duty en 1 heure d'atelier. Le plus important est **qu'ils aient compris comment ça marche** et **qu'ils puissent continuer leur projet chez eux s'ils le souhaitent**. Et surtout (même si ça paraît évident) : qu'ils apprennent et s'amuse ! :)*

C'est quoi Scratch ?

L'objectif de cet atelier est donc d'apprendre la logique de programmation aux enfants.

Pour ce faire vous allez utiliser le logiciel en ligne

SCRATCH



L'usage de blocs de couleurs permet de rendre visuel, et donc plus facilement compréhensible, des concepts abstraits de programmation comme les boucles ou les conditions.

Première chose à faire : s'inscrire sur Scratch !

Avant même de commencer à voir le code, il va falloir s'inscrire sur la plateforme pour pouvoir sauvegarder les jeux et les animations que vont réaliser les enfants.

Pour ce faire, rien de plus simple. Il faut se rendre sur le site : <https://scratch.mit.edu/>

Cliquez ensuite sur le bouton 

Il n'y a plus qu'à suivre les instructions ! **Assurez-vous bien que les enfants aient une adresse mail à renseigner pour pouvoir s'inscrire.**

Scratch, ça marche comment ?

Nous allons voir comment faire pour animer un personnage.



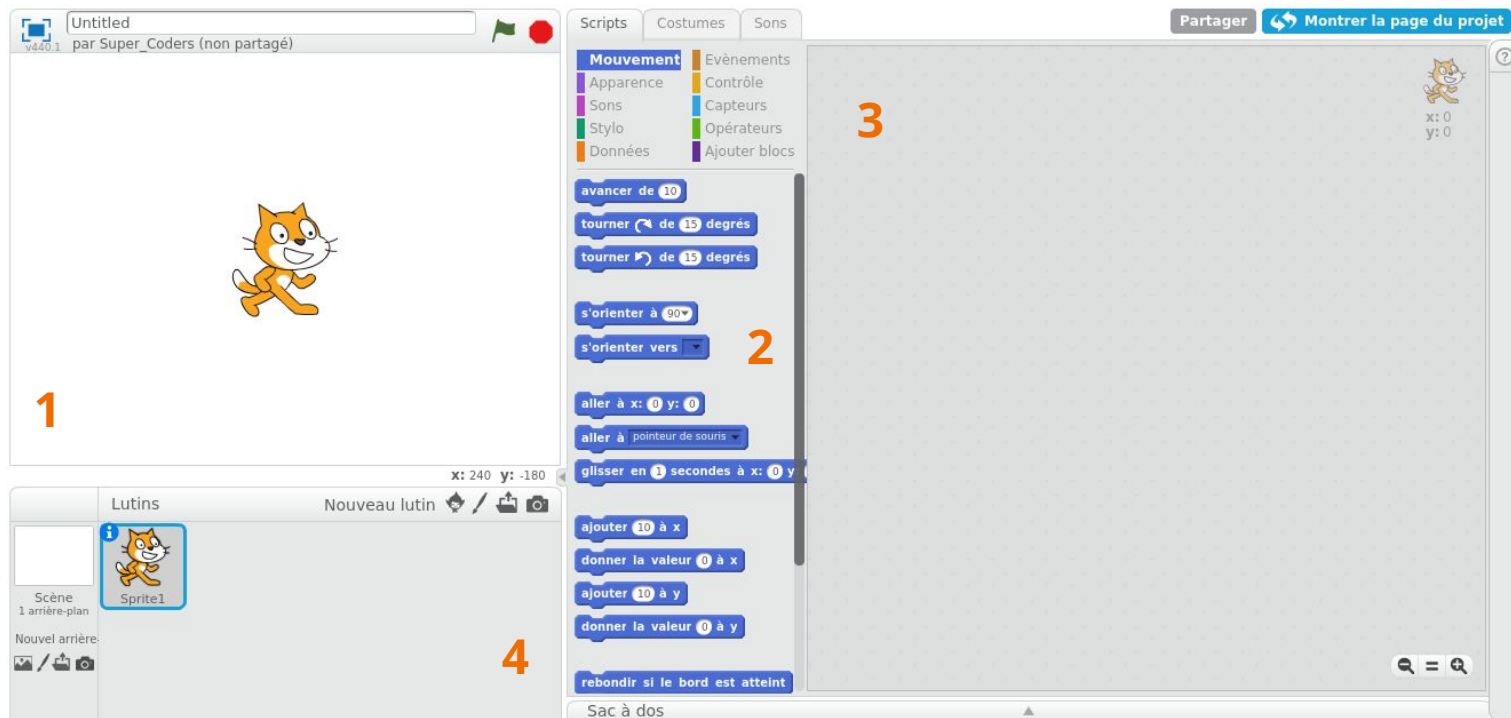
Pour l'instant il ne fait pas grand chose. Il faut lui donner vie !

Et si on essayait de le faire avancer ? Pour cela, il va falloir utiliser des **blocs de code** dans la bibliothèque des **Scripts**.

Un Script c'est quoi ? Vous vous souvenez quand on a fait faire plein d'actions au copain tout à l'heure ? Chaque action est un **script**. En programmation ce sont tous les morceaux de code qui vont permettre à Scratch de bouger.

Se repérer sur une page projet

La page d'un projet ressemble à ceci :



- 1 A gauche la fenêtre qui permet de visualiser le jeu ou l'animation.
- 2 En milieu de page tous les éléments à utiliser pour animer le projet. On y retrouve les scripts, les costumes et les sons.
- 3 A droite la fenêtre où mettre tous les scripts d'animation des personnages et des décors.
- 4 En bas de page se trouve la bibliothèque des **Lutins**. C'est là que s'afficheront tous les personnages utilisés dans le projet.

Animer un personnage

Pour animer un personnage il va falloir lui “commander” des actions, comme on l’a fait avec les petits jeux IRL au début de l’atelier mais cette fois avec du code ! :)

Les différents scripts sont rangés par catégorie :

Mouvement

Données

Opérateurs

Apparence

Évènements

Ajouter blocs

Sons

Contrôle

Stylo

Capteurs

Les catégories que vous allez utiliser le plus souvent sont : mouvement, apparence, évènement, contrôle et capteurs.

Il faudra expliquer à quoi servent ces catégories et les blocs qui les contiennent, mais pour que ça soit plus simple pour vous je vous conseille de donner **une première consigne précise** aux enfants puis de les laisser chercher un peu. Comme pour le jeu que vous avez fait IRL :)

Par exemple : **Quand le drapeau vert est pressé, Scratch avance de 10 pas.**

Il faut ensuite **décomposer la phrase** pour bien comprendre quel bloc de code on va devoir utiliser.

Reprenons donc la consigne :

Quand le drapeau vert est pressé, Scratch avance de 10 pas.

La consigne comporte **deux étapes**. On voit donc qu'on aura besoin de **deux blocs de code**. Mais lesquels ?

Cherchons ensemble dans la bibliothèque de scripts.

Étape 1 : Quand le drapeau vert est pressé...

De quel type de script aurons-nous besoin ? On a besoin qu'un **évènement** (*le drapeau vert pressé*) déclenche une action (*celle de faire avancer le personnage*).

Regardons dans la catégorie **Évènement** si on trouve un bloc qui correspond.

Bingo ! On a la première partie de notre code !



Il suffit maintenant de cliquer sur le bloc, rester appuyer sur la souris et le faire glisser dans la partie **3** de notre fenêtre, celle où on va inscrire toutes les actions de Scratch.

Étape 2 : ... Scratch avance de 10 pas.

On veut faire avancer notre personnage, on comprend donc qu'on va lui faire faire un **mouvement**.

Tiens, il y a une catégorie **mouvement** dans la bibliothèque de scripts. Coïncidence ? Je ne crois pas ! Allons donc jeter un coup d'oeil dans la catégorie pour y trouver le bloc qui nous intéresse.

Super ! On y trouve deux blocs qui peuvent nous servir !

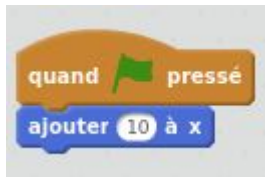


Ce deuxième bloc ne vous rappelle-t-il pas quelque-chose ? ;)

Pour rester cohérent avec la première partie de l'atelier, nous allons utiliser le bloc



Scratch est tellement bien pensé que les différents bloc de code peuvent **s'imbriquer les uns dans les autres**, comme un puzzle ! Ajoutons donc ce deuxième bloc au premier.

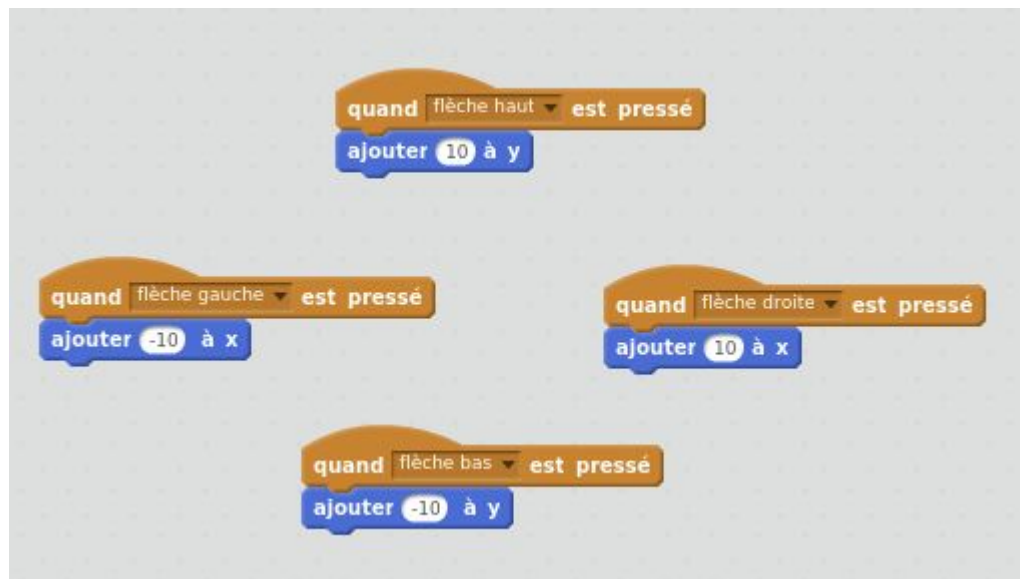


Testons maintenant notre première combinaison pour voir si elle marche. Cliquez sur le drapeau vert pour voir ce qui se passe.

Magie ! Scratch avance comme prévu ! :)

Vous pouvez tester ce même type de code pour faire avancer Scratch avec les flèches du clavier !

En expliquant comme précédemment et en utilisant les blocs suivants, faites faire aux enfants le code pour faire bouger Scratch dans tous les sens !



Les boucles et les conditions avec Scratch

Dans la première partie de l'atelier nous avons pu voir ce qu'était une condition et une boucle. Et si on testait ça avec du code ?

La consigne suivante est un peu plus compliqué que la précédente, mais avec la mise en scène faite au début de l'atelier ces notions sont comprises par les enfants. Leur mise en pratique sera donc plus aisé :)

Commençons à faire un vrai petit jeu. Scratch doit attraper une souris !

Choisissez la souris dans la bibliothèque de Lutins. Maintenant demandons-lui d'exécuter une action.

Nouvelle consigne : **Si la souris est touché, alors Scratch joue le son meow.**

Comme dans l'exercice d'avant, décomposons notre phrase pour bien en saisir toutes les étapes.

On peut repérer trois étapes de code différentes : **Si la souris est touchée, alors Scratch joue le son meow.**

Étape 1 : Si ... alors

Hey ! Mais on connaît ça, c'est une condition ! Une condition permet de **contrôler** une action. Et si on allait voir dans la catégorie **Contrôle** pour voir ce qui s'y trouve ?



Maintenant il ne reste plus qu'à trouver comment compléter ce bloc :)

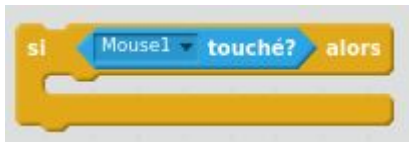
Passons à l'étape 2.

Étape 2 : ... si la souris est touchée ...

Hum, voilà une étape que nous n'avons pas encore vu. On cherche à trouver un bloc de code qui nous permet de **capter** quand notre personnage est touché.

Il me semble avoir repéré une catégorie **Capteurs**. Peut-être y trouverons-nous notre bonheur...

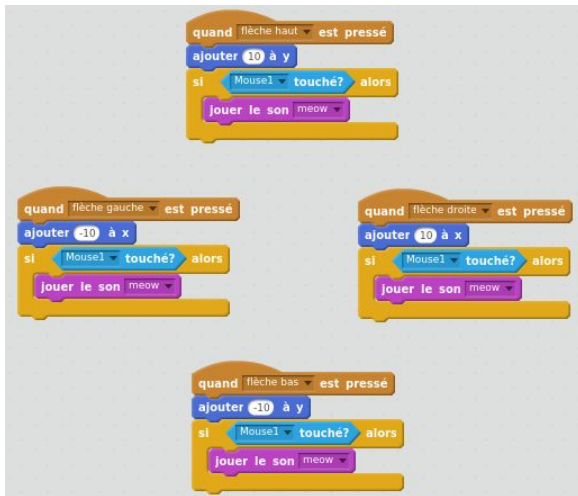
En effet ! Il y a un bloc qui nous permet de dire à l'ordinateur de *capter* lorsque notre personnage est touché. Il suffit de sélectionner "**Mouse1**" dans le menu déroulant. Imbriquons-la ensuite dans notre condition et le tour est joué ! On peut passer à la dernière étape.



Étape 3 : ... Scratch joue le son meow.

Trop facile, on trouvera forcément ce bloc dans la catégorie **son** ! Ajoutons-le dans notre condition, et **imbriquons notre condition sous nos blocs de code “Quand flèche pressée”** pour qu’ils fonctionnent.

On teste ? Si Scratch fait bien meow dès qu’il touche la souris, c’est qu’on a réussi ! :)



Animons maintenant notre souris !

Cliquez sur l'image de la souris dans la partie **4** de votre fenêtre, où se trouve la bibliothèque de Lutins.

Vous verrez votre code précédent s'effacer. Pas de panique c'est normal ! On a encore rien demandé à notre nouveau personnage :)

On corse un peu plus les consignes.

Consigne :

Quand le drapeau vert est pressé, répéter indéfiniment avancer de 10 pas. Il faut que la souris rebondissent si le bord est atteint.

Ajoutons une condition : si Scratch est touché, la souris stop son action.

Pfiou ! Ça m'a l'air compliqué tout ça ! Mais non pas d'inquiétude, si on décompose bien notre consigne comme on l'a fait pour Scratch, on va s'en sortir sans souci !

Comme on dit, commençons par le commencement :

Quand le drapeau vert est pressé, répéter indéfiniment avancer de 10 pas. Faire rebondir la souris si le bord est atteint.

Étape 1 : Quand le drapeau vert est pressé

On connaît déjà ce bloc. Mettons le tout de suite dans la fenêtre qui va recueillir tout notre code.

Étape 2 : ... répéter indéfiniment ...

Tiens, ça ressemble à **une boucle** ça ! Comme les conditions, les boucles permettent de **contrôler** une action. Allons chercher le bloc concerné dans notre catégorie **Contrôle**. Ajoutons le à notre premier bloc.



Étape 3 : ... avancer de 10 pas

Ha ha, il nous est bien familier celui-là ! On ne veut pas forcément que la souris avance sur l'axe de X. Utilisons donc plutôt le bloc **avancer** **de 10** et plaçons le dans notre boucle.



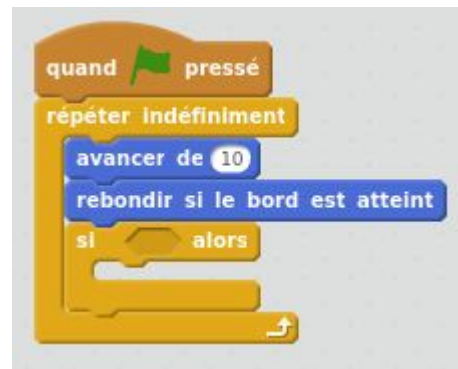
Étape 4 : Faire rebondir la souris si le bord est atteint.

Cette consigne implique un mouvement. Je ne vous redis pas dans quelle catégorie on trouve ce type de bloc, vous la connaissez déjà ;)

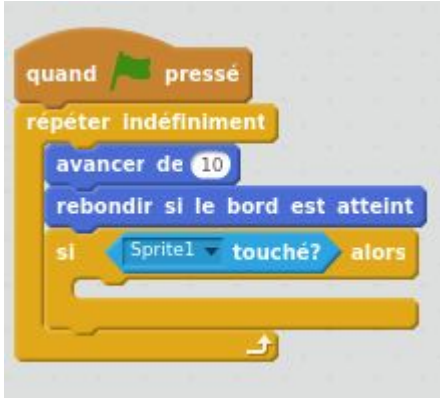


Étape 5 : Si Scratch est touché, la souris stop son action.

Une condition ? Même pas peur ! On sait déjà où aller la chercher :)



Y'a plus qu'à la compléter ! Et en plus on sait déjà quel morceau de bloc on doit utiliser pour que notre souris **capte** quand il touche un autre personnage :)



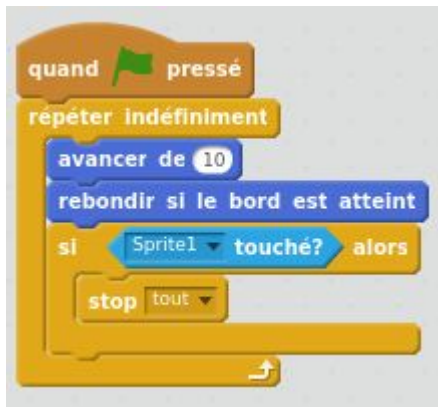
Un dernier petit morceau de code et notre jeu est prêt !

On veut que la souris stop son action au contact de Scratch. On veut donc **contrôler** son mouvement pour l'arrêter.

Pas besoin de vous préciser dans quelle catégorie on va chercher, vous avez déjà deviné !

Oh ! Mais voici un bloc **stop tout** dans la catégorie **Contrôle**. Ajoutons le, et testons ensuite notre code pour voir s'il fonctionne.

Cliquez sur le drapeau vert ... la souris prend vie et se met à parcourir la fenêtre de visualisation du jeu ! Essayez de l'attraper en faisant bouger Scratch avec les flèches de votre clavier. Non seulement la souris s'arrête mais notre petit chat tout content se met à miauler !



Félicitations, vous venez de coder votre tout premier jeu ! :D

PERSONNALISATION DES PROJETS

Maintenant que les enfants ont mis les mains dans le code, laissez libre court à leur imagination !

Gérez bien le temps de votre atelier pour qu'il y ait toujours un petit moment à la fin où les enfants puissent bidouiller sur leur ordinateur les différentes possibilités que propose Scratch.

La librairie de la plateforme contient plusieurs personnages et des décors différents. Qu'ils mettent de la couleur dans ces projets :)

Si vous sentez qu'ils ont du mal à bien comprendre les étapes précédentes de l'atelier prenez plus de temps pour les réaliser, mais insistez bien à la fin pour qu'ils continuent leur jeu à la maison !