

Chapitre 4

Les blocs de programmation

1. Introduction

Scratch est un langage de programmation dit graphique. À la différence des langages textuels, les programmes sont formés par des blocs qui s'assemblent les uns à la suite des autres, les uns dans les autres. Dans ce chapitre, vous allez découvrir les différents blocs utilisables pour créer des programmes. Situés dans l'onglet Scripts, ils sont classés par catégories. Plus d'une centaine de blocs de programmation sont à votre disposition, répartis dans dix catégories aux couleurs différentes, les rendant ainsi plus faciles à identifier.

En plus de ces dix catégories, des blocs liés à des Extensions sont disponibles. Chacune de ces extensions fera l'objet d'un chapitre indépendant avec la présentation de projets spécifiques.

2. Les blocs Mouvement

Les jeux vidéo et les animations comportent différents sprites qui se déplacent et interagissent les uns avec les autres. Certains sont contrôlés par le joueur à l'aide de touches de clavier associées à des blocs **Mouvement**, d'autres sont préprogrammés, d'une manière aléatoire ou non.

Les blocs de la catégorie **Mouvement** servent à placer et déplacer les sprites d'une manière absolue ou relative sur la scène. Ils définissent son orientation et sa position par rapport aux autres sprites.

Dans ce chapitre, vous allez découvrir les spécificités des différents blocs qui composent la catégorie **Mouvement**.

88 Scratch 3 - S'initier à la programmation et à la robotique par le jeu

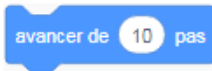
Remarque

Dans le chapitre Techniques pour les jeux vidéo, différents programmes servant à déplacer les sprites seront détaillés.

2.1 Les déplacements relatifs

Les déplacements en fonction de valeurs dites relatives se font soit par rapport à la position du sprite, sans référence aux coordonnées de la scène, soit par rapport à un autre sprite. Lorsque vous déplacez un personnage dans un jeu vidéo, ses déplacements sont des déplacements relatifs.

Trois blocs sont utilisés pour déplacer les sprites dans Scratch d'une manière relative. Ces trois blocs sont pourvus d'une zone de saisie pour spécifier la valeur des pas. Par défaut, celle-ci est de 10 pas. Les pas correspondent à l'unité de la scène.



Le sprite avance de la valeur spécifiée. Pour le faire reculer, il suffit de spécifier une valeur négative : **avancer de - 10 pas**.

Ce bloc est généralement associé à une touche de clavier **quand la touche O est pressée** pour créer une instruction.



Ce bloc modifie la position x du sprite de la valeur spécifiée. Le sprite se déplace horizontalement vers la droite si la valeur spécifiée est positive, vers la gauche si la valeur est négative.

Par exemple, pour un sprite situé à $x = 150$ et $y = 50$, le bloc **ajouter 10 à x** positionne le sprite à $x = 160$ et $y = 50$. La coordonnée y (ordonnée) n'est pas modifiée.

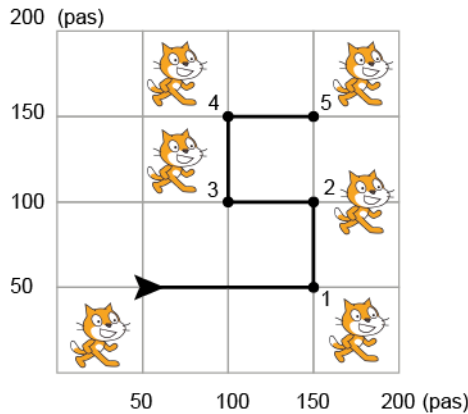


Ce bloc modifie la position y du sprite de la valeur spécifiée. Le sprite se déplace verticalement vers le haut si la valeur spécifiée est positive, vers le bas si la valeur est négative.

Par exemple, pour un sprite situé à $x = 150$ et $y = 50$, le bloc **ajouter 10 à y** positionne le sprite à $x = 150$ et $y = 60$. La coordonnée x (abscisse) n'est pas modifiée.

Exemples d'utilisation

Ce programme est un exemple pour déplacer relativement un sprite sur la scène en modifiant alternativement la valeur de son abscisse et de son ordonnée.



⇒ **quand drapeau vert est cliqué**

⇒ **attendre 1 secondes** // pour insérer un temps d'attente entre l'exécution de chaque bloc mouvement afin que vous puissiez percevoir le déplacement du sprite sur la scène.

⇒ **ajouter 100 à x** // additionne 100 à la valeur x du sprite. Il avance vers la droite (1).

⇒ **attendre 1 secondes**

⇒ **ajouter 50 à y** // additionne 50 à la valeur y du sprite. Il se dirige vers le haut (2).

⇒ **attendre 1 secondes**

⇒ **ajouter - 50 à x** // soustrait 50 à la valeur x du sprite. Il recule (3).

⇒ **attendre 1 secondes**

90 Scratch 3 - S'initier à la programmation et à la robotique par le jeu

⇒ **ajouter 50 à y** // additionne 50 à la valeur y du sprite. Il se dirige vers le haut (4).

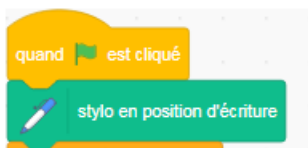
⇒ **attendre 1 secondes**

⇒ **ajouter 50 à x** // additionne 50 à la valeur x du sprite. Il se dirige vers la droite (5).



Remarque

Pour voir sur la scène le tracé du parcours effectué par le sprite, vous pouvez ajouter le bloc **stylo en position d'écriture** (catégorie **Ajouter une extension - Stylo**) au début du programme.

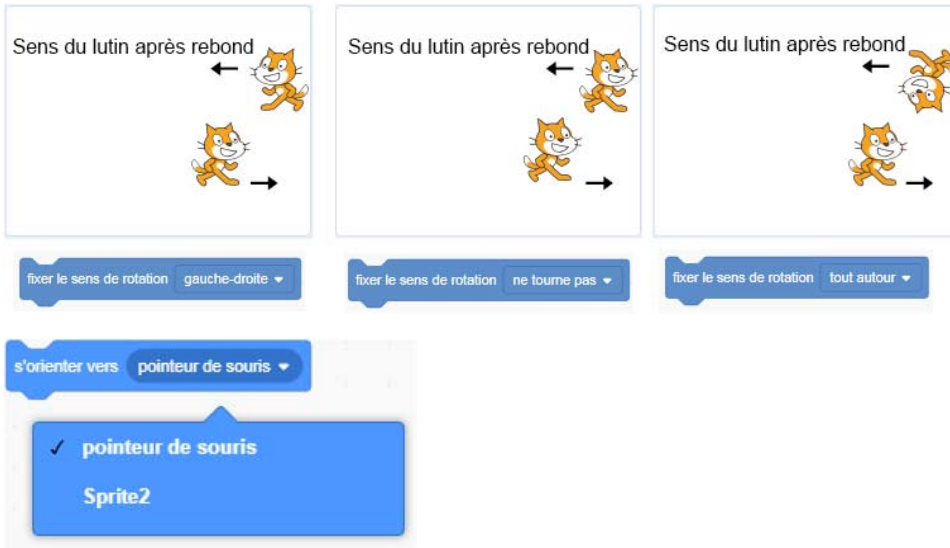


2.2 Orientation et rotation

Plusieurs blocs sont consacrés à l'orientation et à la direction du sprite. Celles-ci peuvent être définies ou s'effectuer en fonction de la souris ou des autres sprites présents.



Ce bloc définit l'orientation prise par le sprite quand il rencontre un obstacle ou l'un des bords de la scène, lorsqu'il s'oriente par rapport à la souris ou à un autre sprite. Trois styles sont sélectionnables en fonction de l'effet recherché. Le style gauche-droite est le plus usité pour un effet plus réaliste lors des changements de direction.



Le sprite se tourne vers le pointeur de la souris ou vers un autre sprite. Tous les sprites utilisés dans le projet sont sélectionnables grâce au menu déroulant.

Exemples

Testez ces deux programmes pour voir comment se comporte le sprite.



Ce bloc est utilisé pour faire tourner le sprite dans le sens horaire, c'est-à-dire dans le sens des aiguilles d'une montre, en fonction d'une valeur modifiable exprimée en degrés.

92 Scratch 3 - S'initier à la programmation et à la robotique par le jeu



Ce bloc est utilisé pour faire tourner le sprite dans le sens antihoraire, c'est-à-dire dans le sens inverse des aiguilles d'une montre, en fonction d'une valeur modifiable exprimée en degrés.



Ce bloc est utilisé pour orienter le sprite afin qu'il se déplace dans la bonne direction. En sélectionnant la zone de saisie, un cercle faisant penser à un cadran d'horloge, ou à un rapporteur faisant 360 degrés, s'affiche. Il suffit de modifier la position de la flèche à l'aide de la souris pour déterminer l'orientation :

- vers le haut (0°) ;
- vers le bas (180°) ;
- vers la droite (90°) ;
- vers la gauche (-90°).

Ces quatre angles sont utilisés dans les programmes pour déplacer le sprite sur la scène. D'autres valeurs peuvent être spécifiées, notamment pour faire rebondir des balles.