Chapitre 5
Les blocs de programmation

1. Introduction

Scratch est un langage de programmation dit visuel. À la différence des langages textuels, les programmes sont élaborés grâce à 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. Les blocs spécifiques pour contrôler les GPIO du Raspberry Pi seront détaillés dans les prochains chapitres.

Situés dans l'onglet **Code**, les blocs classiques sont classés par catégories et par couleurs. Plus d'une centaine de blocs de programmation sont à votre disposition.

D'autres doivent être ajoutés en sélectionnant **Ajouter une extension**.

Ce chapitre n'est pas à lire d'une manière linéaire. Je vous conseille de venir vous y référer lorsque vous vous posez une question sur les caractéristiques d'un bloc et son utilisation.

Nous allons dans un premier temps découvrir les bocs "classiques", puis nous nous intéresserons à certaines extensions.

2. Les blocs Mouvement

Les jeux vidéo et les animations sont composés d'éléments graphiques, appelés sprites (lutins) qui se déplacent sur l'écran. Certains se déplacent, suite à l'action du joueur. D'autres sont indépendants c'est-à-dire que leurs déplacements sont préprogrammés.

Les déplacements sont gérés grâce aux blocs situés dans la catégorie Mouvement.

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. Lorsqu'un joueur déplace lui-même son personnage dans un jeu, les déplacements sont des déplacements relatifs.

Trois blocs sont utilisés pour déplacer les sprites d'une manière relative. Ils sont pourvus d'une zone de saisie pour spécifier la valeur des pas (taille des déplacements). Par défaut, celle-ci est de 10 pas. Les pas sont l'unité de la scène.

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

Pour créer une instruction de déplacement, ce bloc est généralement associé à une touche de clavier de type **quand la touche () est pressée**.

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, et 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. Et 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.

2.2 Orientation et Rotation

Un sprite peut s'orienter et se diriger dans toutes les directions sur la scène.

Ce bloc définit l'orientation prise par le 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.

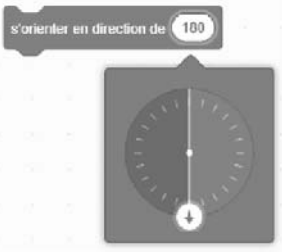
Le sprite tourne de 15° dans le sens horaire, c'est-à-dire dans le sens des aiguilles d'une montre. La valeur, exprimée en degrés, est modifiable.

Le sprite tourne de 15° dans le sens anti horaire, c'est-à-dire dans le sens contraire des aiguilles d'une montre. La valeur, exprimée en degrés est modifiable.

Ce bloc définit la direction prise sur la scène par le sprite lorsqu'il se déplace. 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 fixer l'orientation :

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

Pour se diriger vers le haut



Pour se diriger vers le bas

Pour se diriger vers la gauche

Pour se diriger vers la droite

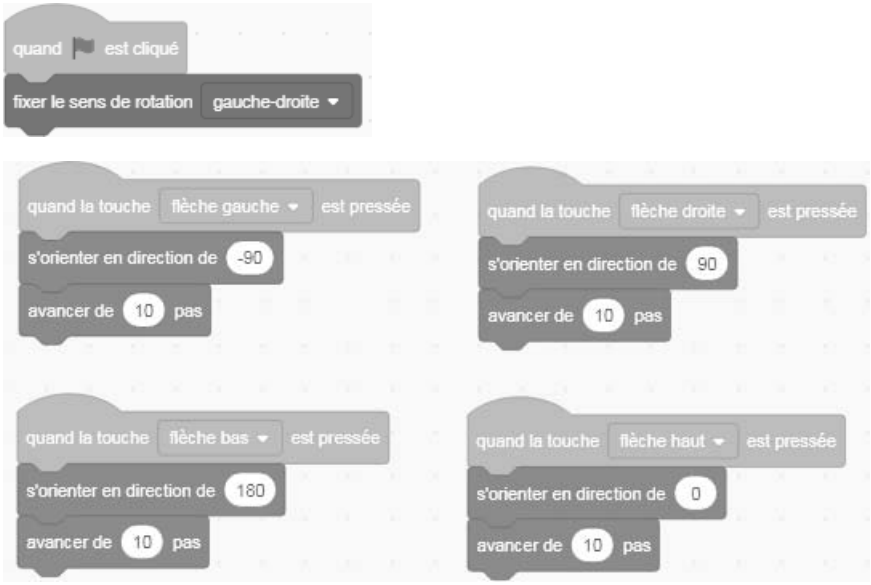
Le sprite pivote et se positionne dans la direction spécifiée. 360 angles différents peuvent être définis.

Remarque

Pour éviter que le sprite ne se retrouve la tête à l'envers lors des changements de direction, il faut penser à **fixer le sens de rotation gauche-droite**.

Exemple de programme de déplacement

Voici un exemple pour déplacer un sprite dans les quatre directions en utilisant les flèches directionnelles.



2.3 Les déplacements absolus

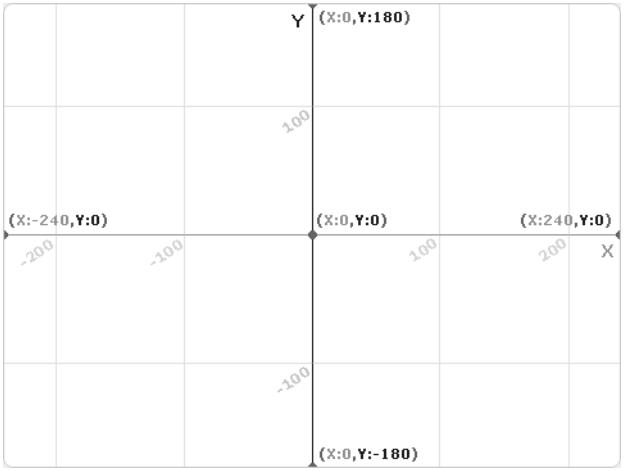
La scène sur laquelle se déplacent les sprites est un rectangle de 480 pas sur 360 pas.

La scène correspond à un repère orthonormé.

- L'axe des abscisses (x) est gradué de -240 à +240.
- L'axe des ordonnées (y) est gradué de -180 à +180.

Le centre de la scène correspond aux coordonnées x = 0 (abscisse) et y = 0 (ordonnée).

Tous les points situés vers le haut et vers la droite correspondent à des coordonnées positives.
Tous les points situés vers le bas et vers la gauche correspondent à des coordonnées négatives.



Quatre blocs servent à déplacer le sprite sur la grille de la scène en fonction de valeurs absolues, c'est-à-dire en fonction de coordonnées x et y.

Le sprite se positionne au niveau des coordonnées spécifiées. Le déplacement s'effectue rapidement : le sprite disparaît et réapparaît à l'endroit spécifié.

Remarque

Au démarrage d'un programme, que ce soit un jeu ou une animation, il peut être important de positionner les sprites à des endroits bien définis. Par exemple, lors de la création d'un jeu de type labyrinthe, le sprite joué sera positionné à l'entrée du labyrinthe au démarrage du jeu, et pourra y être renvoyé en cas d'échec lors de ses déplacements au sein du labyrinthe.

Quand l'exécution du programme commence, le sprite se positionne aux coordonnées x = 100 et y = 40.

Un sprite peut également être déplacé en modifiant indépendamment la valeur x ou la valeur y de ses coordonnées. En modifiant la valeur x, le sprite se déplace horizontalement. En modifiant la valeur y, il se déplace verticalement.