

# Gérer des conditions, des capteurs et des angles

## 17 Utiliser des conditions et des capteurs

**Objectif**  
Comparer deux techniques de gestion d'événements.

### Je découvre

Voici 2 techniques pour gérer un événement, comme « presser la flèche droite ».

**Technique 1 (simple)**

Elle utilise une brique de la catégorie « Événements » et lui emboîte des briques d'« actions ». Elle est facile à utiliser mais ne gère qu'un nombre limité d'événements.

**Technique 2 (avancée)**

Elle teste sans arrêt si l'événement se produit, en utilisant une boucle « répéter indéfiniment », une **condition** et un **capteur**. Elle est plus réactive que la technique 1 et gère plus d'événements.

```

quand flèche droite est pressée
  ajouter 10 à x
    
```

```

quand est cliqué
  répéter indéfiniment
    si touche flèche droite pressée alors
      ajouter 10 à x
    
```

### À toi de jouer !

#### Projet 23 Course sous-marine



1. Crée ce projet avec les deux plongeurs que tu places l'un en dessous de l'autre.

2. a. Applique le script de la technique 1 pour le plongeur violet, puis le script de la technique 2 pour le plongeur orange. Qui arrive le premier ?

b. Pourquoi ?

3. a. On veut tester sans arrêt si le bord est touché. Mais ce script ne fonctionne pas. Pourquoi ?

```

Au lancement du projet :
  Si le bord est touché alors :
    Afficher « Touché ! ».
    
```

b. Crée le script corrigé pour chaque plongeur.

4. Teste et enregistre ton projet.

#### Projet 24 début Sauts de grenouille



1. Crée ce projet en plaçant le lutin « Frog » sur un fond sous-marin.

2. a. Complète l'algorithme en suivant ces instructions :

- Sans arrêt :
- La grenouille avance de 10.
  - Elle rebondit si elle touche le bord.
  - Son poids la fait descendre de 10.

Au lancement du projet :

b. Crée le script associé.

3. Ajoute le script pour faire monter la grenouille de 20 si **espace** est pressée.

4. Teste et enregistre ton projet.

## 18 Toujours plus de conditions

**Objectif**  
Manipuler des conditions avancées.

### Je découvre

Voici comment utiliser des conditions dans des situations particulières :

• Faire avancer un lutin de 10 si **A** est pressée, sinon reculer de 10 :

```

si touche A pressée? alors
  avancer de 10
sinon
  avancer de -10
    
```

La condition.  
Action si la condition est réalisée.  
Action si la condition n'est pas réalisée.

• Stopper le jeu si le lutin touche le bord ou s'il touche du jaune :

```

si bord touché? ou couleur touchée? alors
  stop tout
    
```

Deux conditions : soit l'une ou soit l'autre.

• Faire avancer le lutin si le bord est non touché :

```

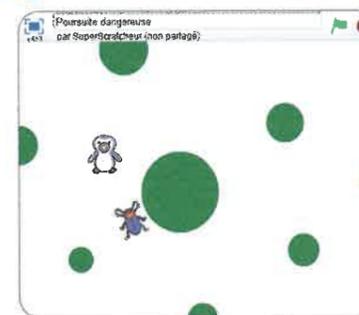
si non bord touché? alors
  avancer de 10
    
```

On utilise aussi la brique **et** pour tester si 2 conditions sont réalisées.



### À toi de jouer !

#### Projet 25 Poursuite dangereuse



Le pingouin doit échapper au scarabée qui le poursuit sans toucher les obstacles verts.

1. Crée ce projet avec les lutins « Penguin1 » et « Beetle ».

2. Dessine, sur la scène, des disques verts de taille différente et suffisamment espacés comme ci-dessus.

3. Crée le script qui déplace indéfiniment le pingouin à la position de la souris.

4. Crée un autre script tel que, quand le pingouin est cliqué :

- a. Il envoie le message « C'est parti ! »
- b. Il teste indéfiniment s'il touche « Beetle » ou la couleur verte. Si oui : le jeu s'arrête.

5. a. Complète cet algorithme en suivant ces instructions :

Quand « Beetle » reçoit le message « C'est parti ! », il répète sans arrêt :

- Il avance de 10.
- S'il touche du vert, il recule de 10, tourne à gauche de 60° et avance de 10.
- Sinon, il s'oriente vers le pingouin.

Quand « Beetle » reçoit « C'est parti ! » :

Répéter indéfiniment :

Si

Sinon :

b. Crée le script associé à cet algorithme dans Scratch.

6. Teste et enregistre ton projet.