

- ❖ Tutoriel de **développeur**
- ❖ Niveau **facile**



Déplacer un personnage 2D dans toutes les directions

Rappels :

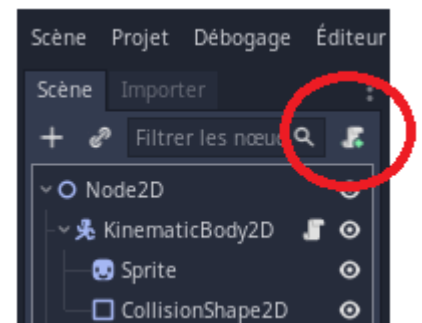
- Ce tutoriel vous permet de cocher une case d'expérience dans votre **carnet de progression** en tant que **développeur**.
- Le vocabulaire propre au moteur GODOT est écrit en *italique*.

Objectifs du tutoriel :

- Faire se déplacer une image (*sprite*), qui représente un personnage dans un environnement en 2D.
- Le joueur, en appuyant sur les touches fléchées de son clavier, fera bouger le *sprite* dans toutes les directions (comme un jeu A-RPG de style Zelda)

Etapas à suivre :

1. Après avoir lancé un nouveau projet sur GODOT Engine, créez d'abord une *Scène2D* (une scène est un « niveau » de jeu vidéo).
2. Ajoutez au *Node2D* (en haut à gauche de l'écran) un nœud enfant *KinematicBody2D* (utilisez le clic droit sur le *Node2D*).
3. A ce *KinematicBody2D*, ajoutez comme nœud enfant une *Sprite* (une image du personnage, par défaut vous pouvez prendre l'icône de Godot *icon.png* (en bas à gauche, faites-la glisser depuis le Système de fichiers vers la propriété *Texture* de la *Sprite* en haut à droite).
4. Toujours à ce *KinematicBody2D*, ajoutez comme nœud enfant une *CollisionShape2D*. Dans la propriété *Shape* de la *CollisionShape2D*, sélectionnez "Nouveau *RectangleShape2D*" et redimensionnez le rectangle pour remplir l'image de la *Sprite*.
5. Cliquez sur le nœud *KinematicBody2D*, puis sur l'icône « attacher un nouveau script » (le parchemin au-dessus des nœuds avec un petit « + » vert) voir image à droite →
6. Dans le *script*, écrivez le code suivant :



```
extends KinematicBody2D
```

```
var speed = 250
var velocity = Vector2()
```

```
func get_input():
    velocity = Vector2()
    if Input.is_action_pressed('ui_right'):
        velocity.x += 1
    if Input.is_action_pressed('ui_left'):
        velocity.x -= 1
    if Input.is_action_pressed('ui_down'):
        velocity.y += 1
    if Input.is_action_pressed('ui_up'):
        velocity.y -= 1
    velocity = velocity.normalized() * speed
```

```
func _physics_process(delta):
    get_input()
    move_and_collide(velocity * delta)
```

7. Jouez la scène en cliquant sur l'icône « lecture » en haut à droite de l'écran. Votre *sprite* devrait se déplacer grâce aux touches fléchées du clavier !
8. Apprenons tout de même des choses, voici quelques explications sur le vocabulaire du *GDScript*, le langage informatique utilisé pour coder GODOT :

<i>KinematicBody2D</i>	est un corps qui va exister dans un espace en 2D (ici, c'est votre personnage, mais cela convient aussi aux ennemis ou personnages non-joueur).
<i>Var</i>	« variable », une variable est une information que vous donnez à votre code.
<i>Speed</i>	« vitesse » calculée en pixel par seconde.
<i>Vector2</i>	déplacement dans un environnement 2D
<i>Func</i>	« fonction », une fonction est une mission que vous donnez à votre code
<i>If</i>	« si », une condition que vous donnez au code, ses effet s'appliquent si cette condition est remplie
<i>Velocity.x</i>	déplacement sur l'axe horizontal
<i>Velocity.y</i>	déplacement sur l'axe vertical
<i>Input</i>	pression d'un bouton par le joueur (touche du clavier, d'une manette, clique de souris...)

9. Quelques explications sur le contenu de votre code :

```

extends KinematicBody2D

var speed = 250
var velocity = Vector2()

func get_input():
    velocity = Vector2()
    if Input.is_action_pressed('ui_right'):
        velocity.x += 1
    if Input.is_action_pressed('ui_left'):
        velocity.x -= 1
    if Input.is_action_pressed('ui_down'):
        velocity.y += 1
    if Input.is_action_pressed('ui_up'):
        velocity.y -= 1
    velocity = velocity.normalized() * speed

func _physics_process(delta):
    get_input()
    move_and_collide(velocity * delta)

```

Indique que ce code utilise un nœud KinematicBody2D comme référence.

Ces variables indiquent la vitesse de l'objet (ici mon personnage) et qu'il va se déplacer en 2D.

Cette fonction gère les mouvements selon les touches du clavier (Input) utilisées par le joueur.

Cette fonction fait que le code calcule la position de l'objet à chaque frame (plusieurs dizaines de fois par secondes)

Les espaces laissés ici sont écrits avec la touche TAB du clavier. On parle d'*indentation* : la ligne indentée est un « enfant » de la ligne supérieure, elle appartient à sa *fonction* et n'existe qu'en elle.

10. Bravo, vous avez terminé ce tutoriel et gagné un point d'expérience de **développeur** !