

Kinematic Character 2D – Explication complète

1. Qu'est-ce qu'un Kinematic Character ?

Un **Kinematic Character** est un type de personnage dans Godot (CharacterBody2D / KinematicBody2D) dont **le mouvement est entièrement contrôlé par le code**.

Il ne réagit pas automatiquement aux forces physiques :

- il ne tombe pas tout seul,
- il ne saute pas tout seul,
- il ne rebondit pas,
- il n'est pas poussé par d'autres objets.

Le moteur physique fait UNE chose automatiquement :

il empêche ton personnage de traverser les murs et le sol.

2. Pourquoi on utilise ça ?

Parce que c'est parfait pour : - les jeux de plateforme (Mario, Celeste), - les jeux d'action, - les jeux où la précision du mouvement est importante.

Avec un personnage cinématique : - tu contrôles exactement comment il bouge, - tu règles toi-même la gravité, - tu détermènes la vitesse, - tu choisis comment les collisions réagissent.

3. Comment ça fonctionne dans Godot

Le mouvement se gère dans `_physics_process(delta)` afin d'être synchronisé avec la physique.

Tu manipules un vecteur de vitesse :

```
var velocity = Vector2()
```

Déplacement horizontal

```
if Input.is_action_pressed("ui_left"):
    velocity.x = -200
elif Input.is_action_pressed("ui_right"):
    velocity.x = 200
else:
    velocity.x = 0
```

Gravité

```
velocity.y += GRAVITY * delta
```

Saut

```
if is_on_floor() and Input.is_action_just_pressed("jump"):
    velocity.y = -JUMP_FORCE
```

Application du mouvement

```
velocity = move_and_slide(velocity)
```

`move_and_slide()` : - empêche de traverser les murs, - empêche d'entrer dans le sol, - fait glisser le personnage le long des surfaces.

4. Ce que Godot NE fait PAS automatiquement

- il ne modifie pas la vitesse si tu frappes un mur (sauf pour bloquer),
- il ne crée pas de rebond,
- il ne gère pas d'accélération automatique,
- il ne simule pas le poids ou la masse.

Tu dois coder toi-même : - la gravité, - le saut, - les arrêts, - le comportement sur les pentes, - la friction si nécessaire, - les glissades spéciales.

5. Scène typique

```
CharacterBody2D
    CollisionShape2D
    Sprite2D / AnimatedSprite2D
```

Règle importante

Ne jamais scaler un `CollisionShape2D`.

Toujours modifier la forme directement (rayon, taille, etc.).

6. Résumé global

- Tu contrôles 100% du mouvement.
 - Godot ne fait que bloquer les collisions.
 - Parfait pour obtenir un mouvement ultra précis.
 - Utilisé dans presque tous les platformers 2D modernes.
-

7. En une phrase

Un Kinematic Character, c'est un personnage que tu déplaces toi-même avec ton code, et Godot empêche juste qu'il traverse les obstacles.