Pract- Create simple enemy objects that move automatically.

Open Godot and create a new scene.

- Click **Scene → New Scene**.

Create the Enemy node

- Add a **Node2D** as the root. Rename it to Enemy.

Add a sprite

- With Enemy selected click **+** and add **Sprite2D**.
- In the Inspector, load a texture (an enemy image) into Texture.
- Make sure the node name is exactly Sprite2D (the script will use that name).

Add a collision shape (so it can touch player later)

- Add **CollisionShape2D** as a child of Enemy.
- In the Inspector set Shape → **New RectangleShape2D** (resize it to cover the sprite).

Save the scene

- Save as Enemy.tscn.

Attach a script to Enemy

- With Enemy selected click **Attach Script** and choose GDScript. Paste this script into it:

```
# Enemy.gd  (for Godot 4)
extends Node2D

@export var speed: float = 100.0    # how fast the enemy moves (pixels/sec)
@export var left_x: float = -200.0  # left patrol X (you can set these in the Inspector)
@export var right_x: float = 200.0  # right patrol X
var dir: int = 1                    # 1 = moving right, -1 = moving left

func _ready() -> void:
    # make sure starting position is inside limits
```

```
    position.x = clamp(position.x, left_x, right_x)

func _physics_process(delta: float) -> void:
    # move
    position.x += dir * speed * delta

    # when we reach the right limit, go left; when we reach left limit, go right
    if position.x >= right_x:
        position.x = right_x
        dir = -1
    elif position.x <= left_x:
        position.x = left_x
        dir = 1

    # flip the sprite so it faces the direction it's moving
    if has_node("Sprite2D"):
        $Sprite2D.flip_h = dir < 0
```

Set patrol points & speed in the Inspector

- Select the Enemy node in the scene. You will see speed, left_x, and right_x in the Inspector (because we exported them).
- Example: if your main scene width is 800 px and enemy starts at x=400, you might set left_x = 200 and right_x = 600.

Test it

Press the Play button (F5). Your enemy should walk back and forth between the left and right X values and flip its sprite when changing direction.