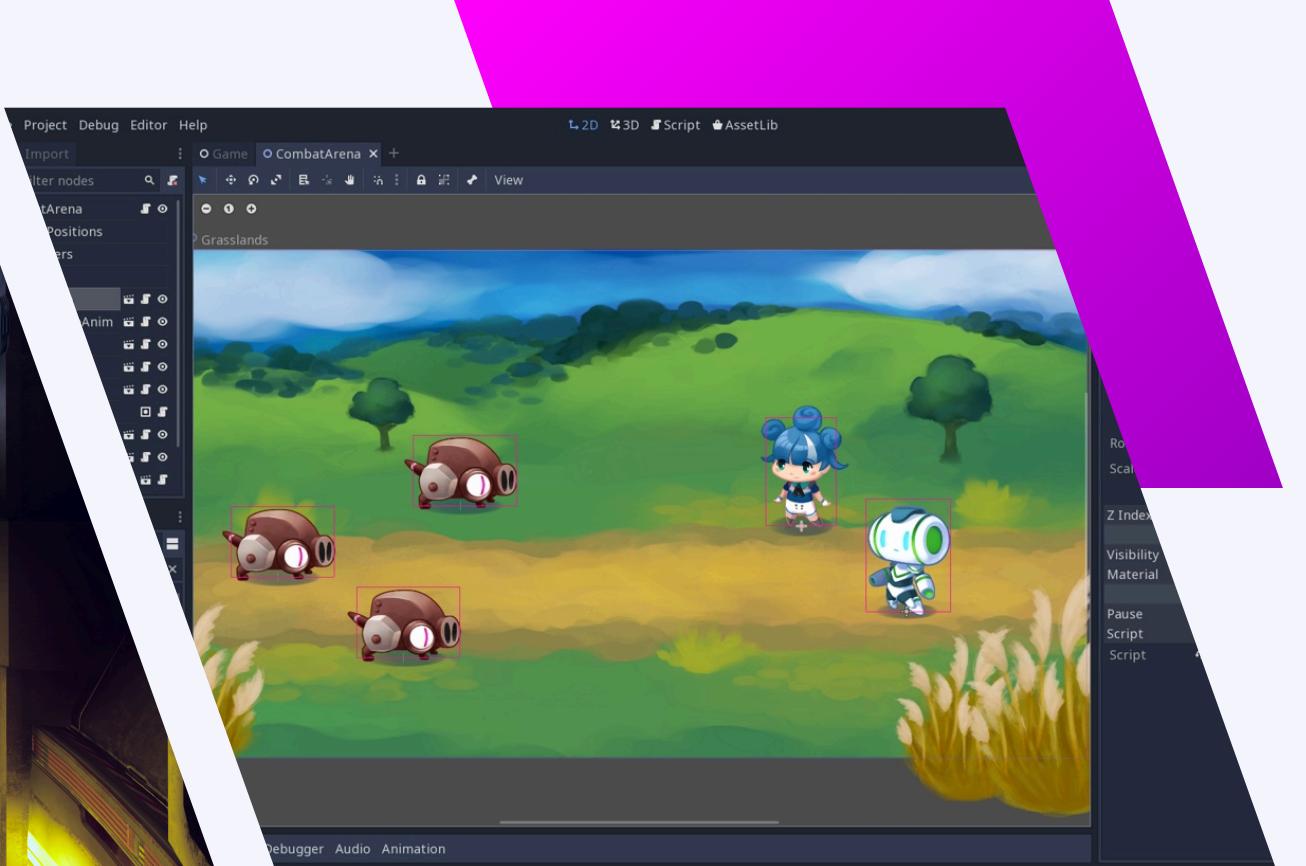
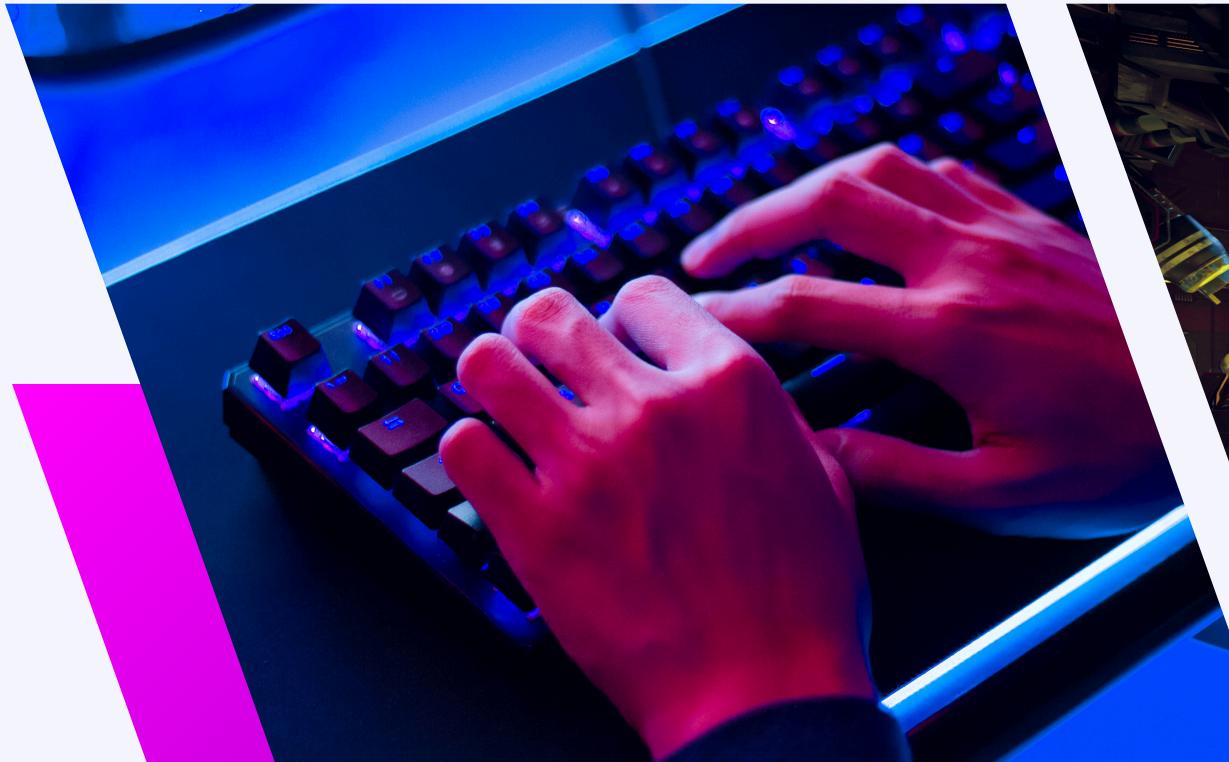
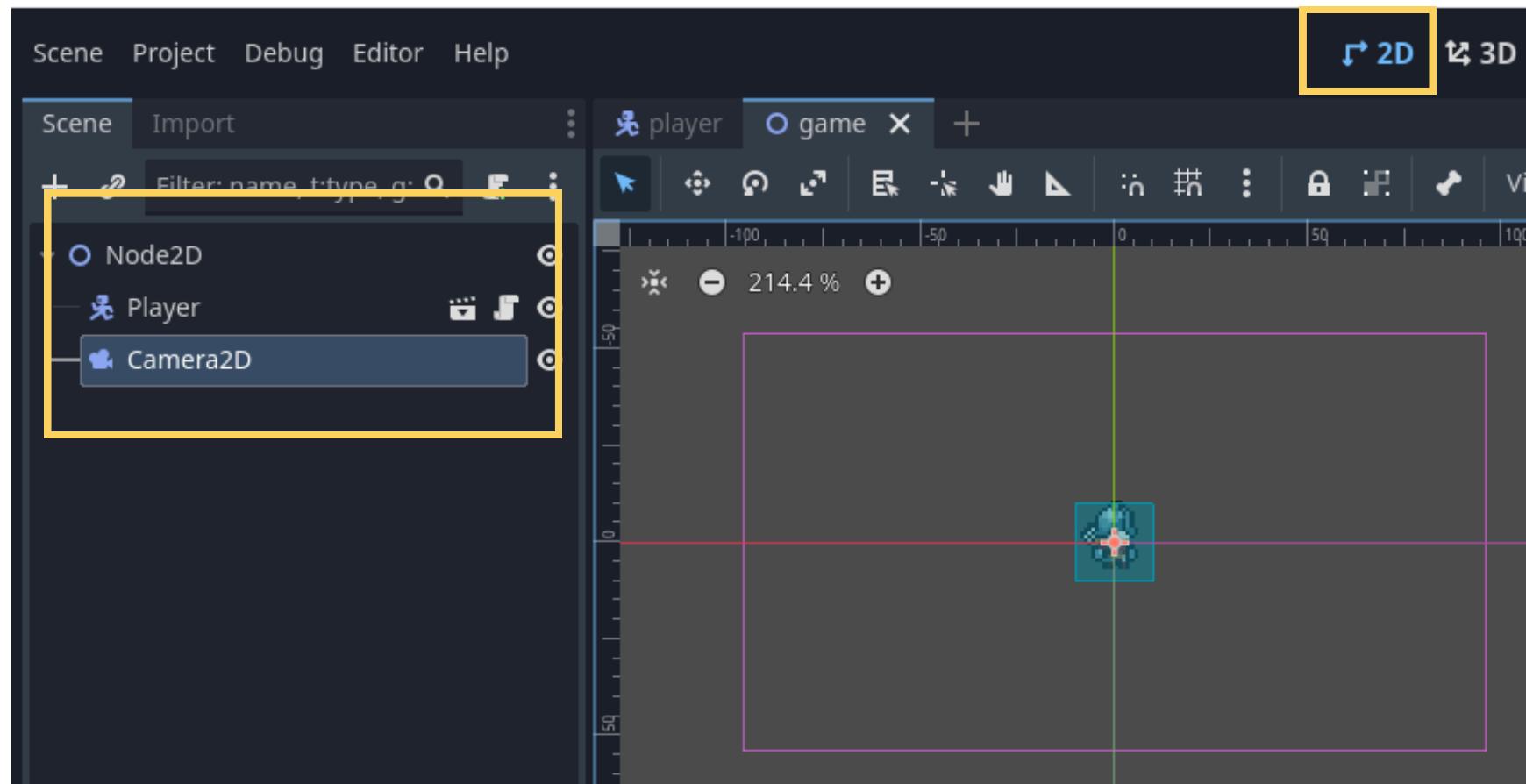


GAME DEVELOPER

College of Creative Design and Entertainment Technology



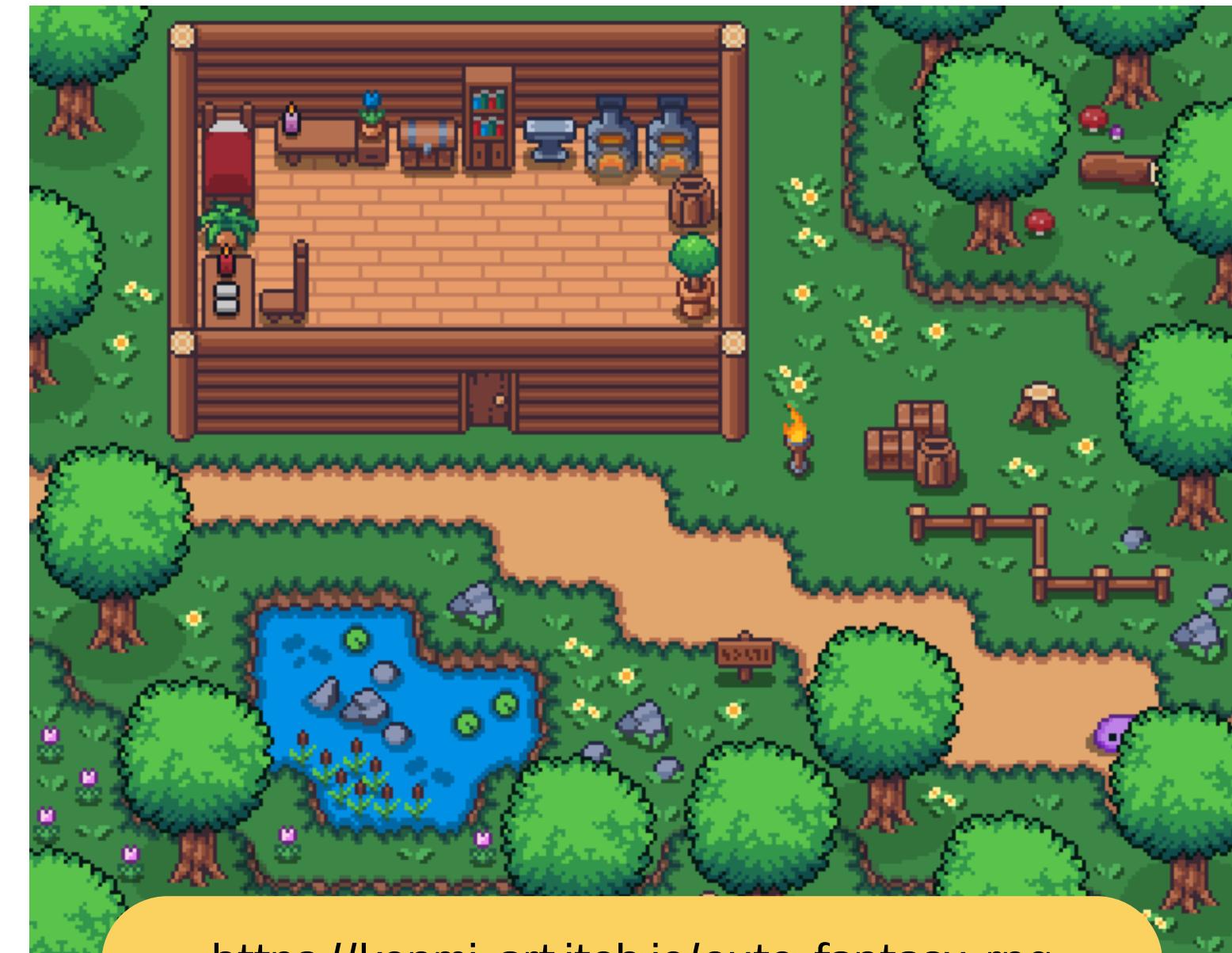
การพัฒนาเกมด้วยโปรแกรม Godot Engine



หัวข้อหลักของบทเรียนนี้จะเป็นเรื่องการจัดการส่วนของ Level Design ของเกมเรา กลับไปที่ [game.tscn](#) ส่วนของ 2D

ขั้นตอนที่ 1 การจัดการ TileSet

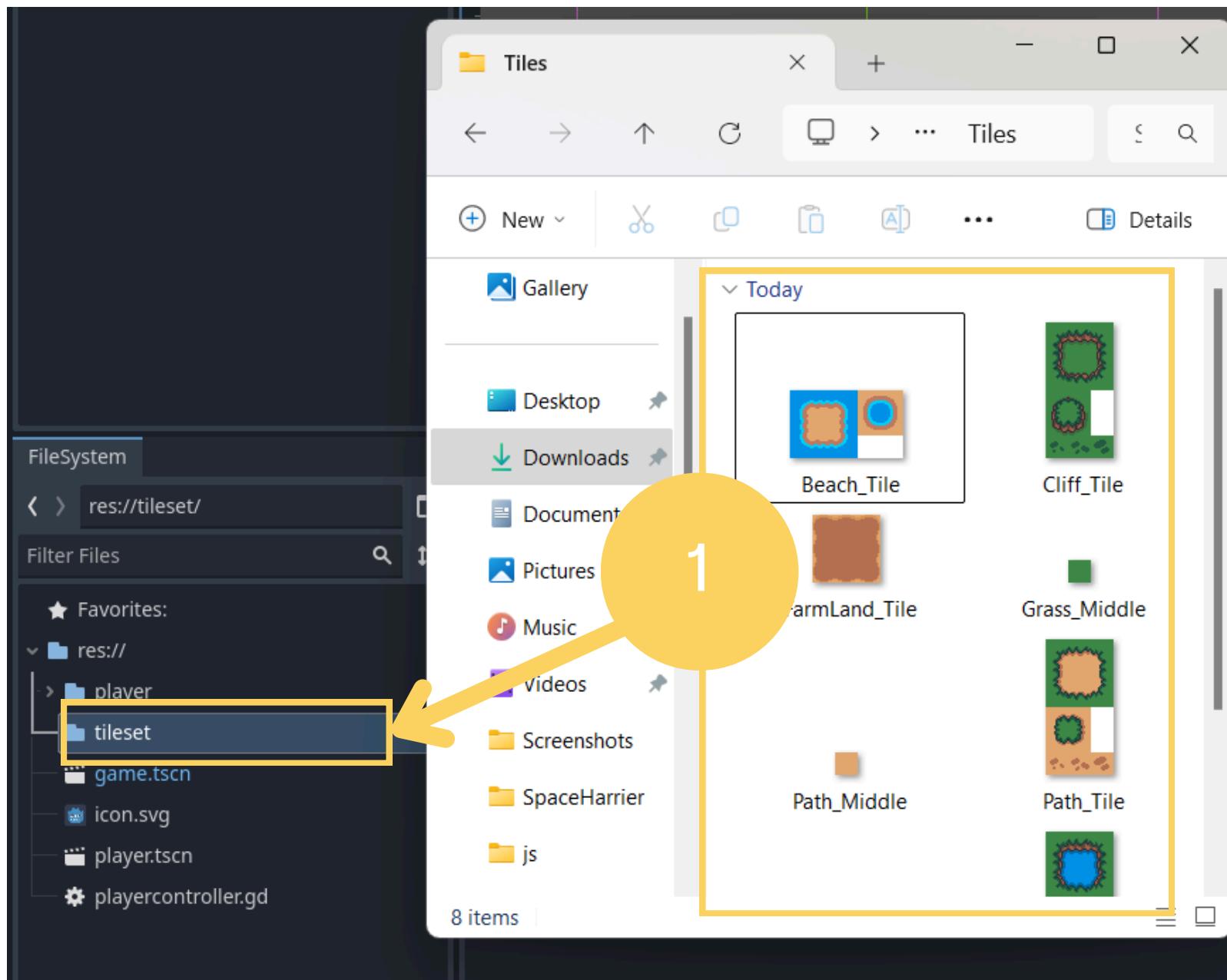
TileSet คือชุดของภาพหรือวัตถุกราฟิกที่ใช้ในการสร้างจากเกมแบบเป็นตาราง โดยเฉพาะในเกมแนว 2D (Top-down, Platformer, Puzzle) ใน Godot Engine, TileSet ใช้ร่วมกับ TileMap เพื่อสร้างจากขนาดใหญ่ได้อย่างมีประสิทธิภาพ โดยไม่ต้องวาดหรือวางแผนดูๆ กัน



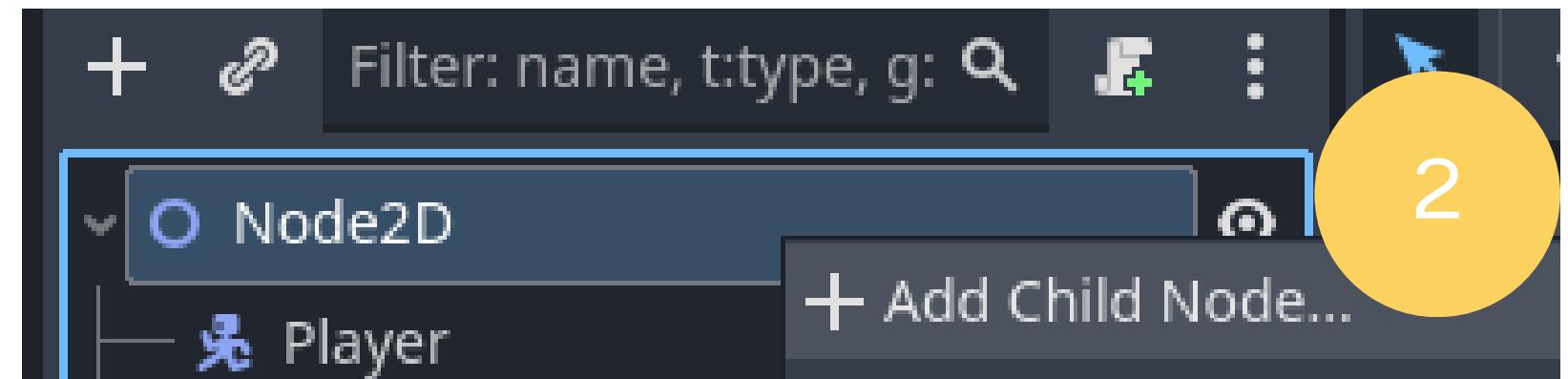
<https://kenmi-art.itch.io/cute-fantasy-rpg>

เราจะใช้ TileSet Assets ตัวอย่างนี้ซึ่งทุกคนสามารถดาวน์โหลดได้ที่:

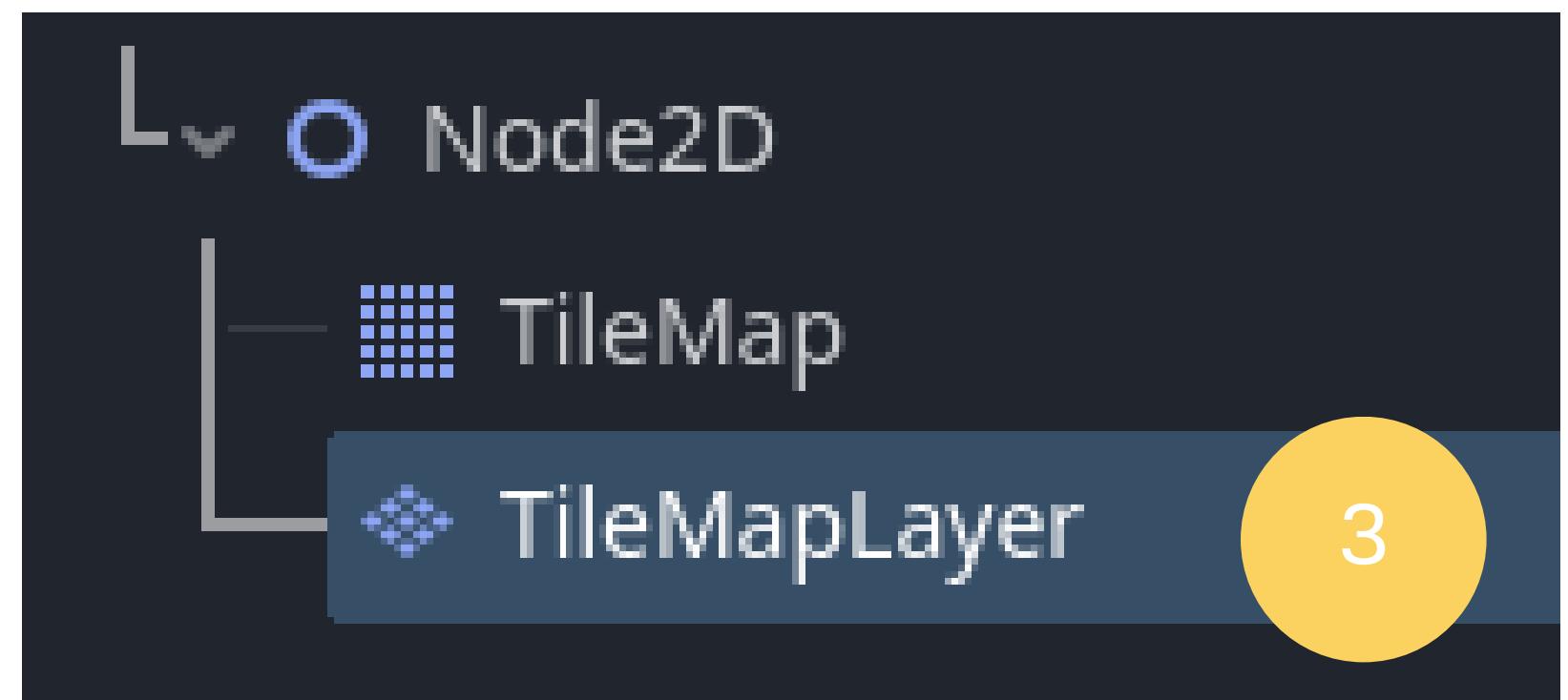
https://drive.google.com/file/d/1DOYPeAJdk1A4moB5nmz-woW0e8co07WI/view?usp=drive_link

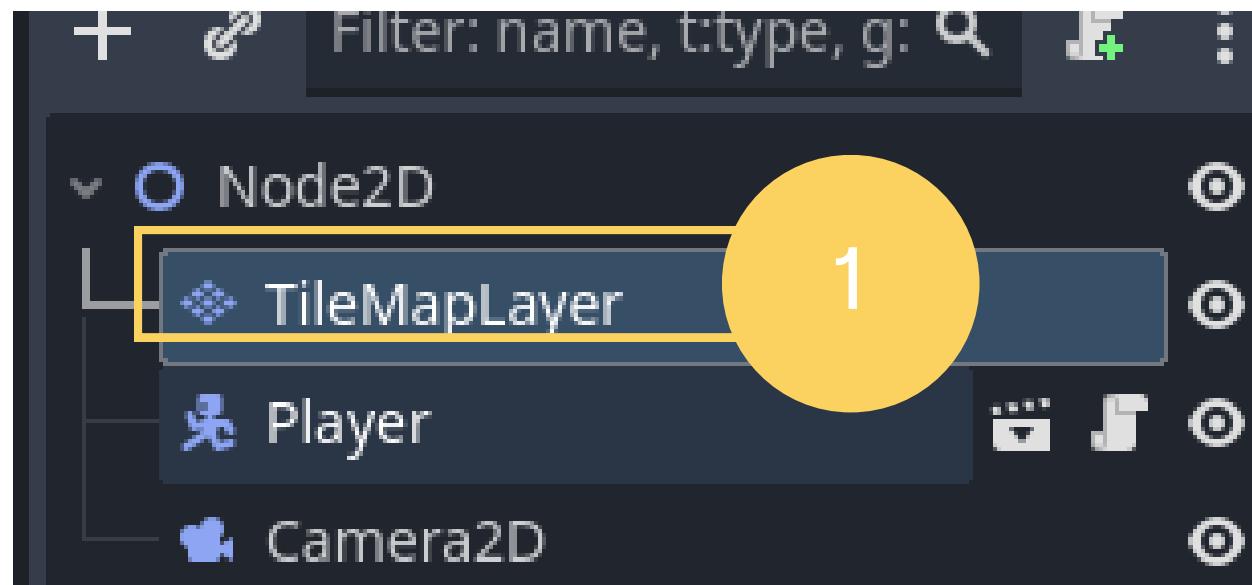


สร้าง folder ใหม่ใน **FileSystem** ว่า **tileset** หลังจากนั้นให้เราไฟล์กันหนดที่ Unzip แล้วลากไปปล่อยใน tileset เพื่อเป็นการจัดการไฟล์ที่เราต้องใช้สร้างระบบ Level Design ของเราระบบ Paint Texture ปกติ

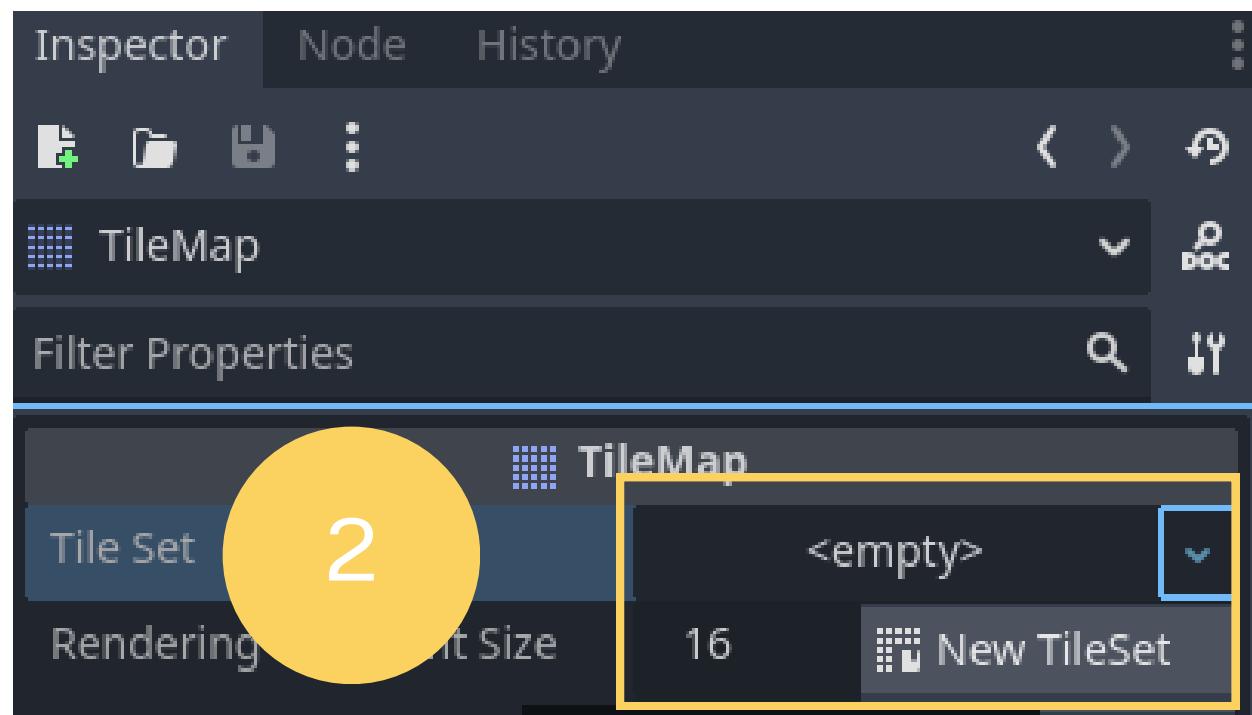


คลิกขวาที่ Node2D ในจาก **game.tscn** หลังจากนั้นเลือก **+ Add Child Node...** แล้วทำการเลือก **TileMapLayer**

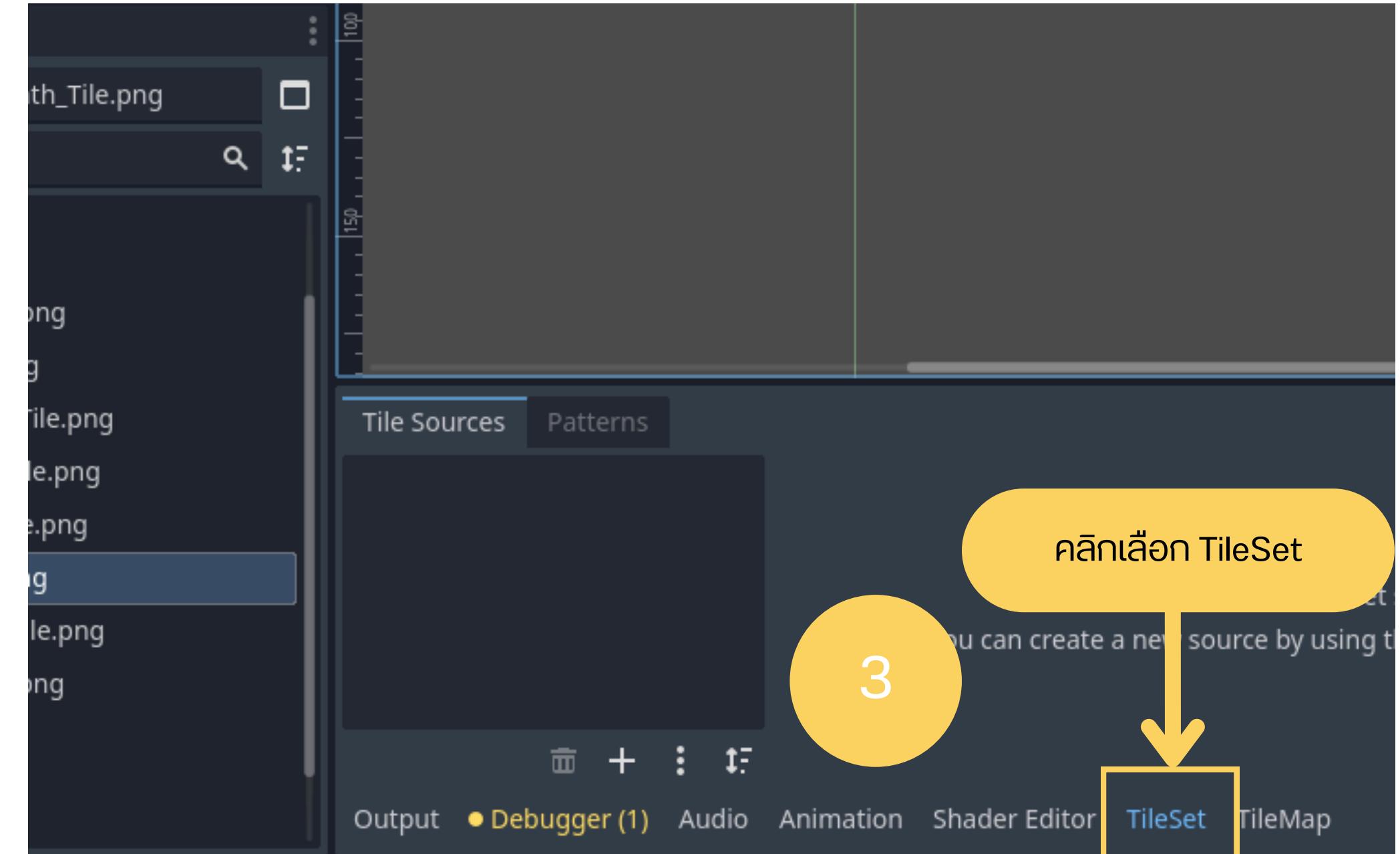




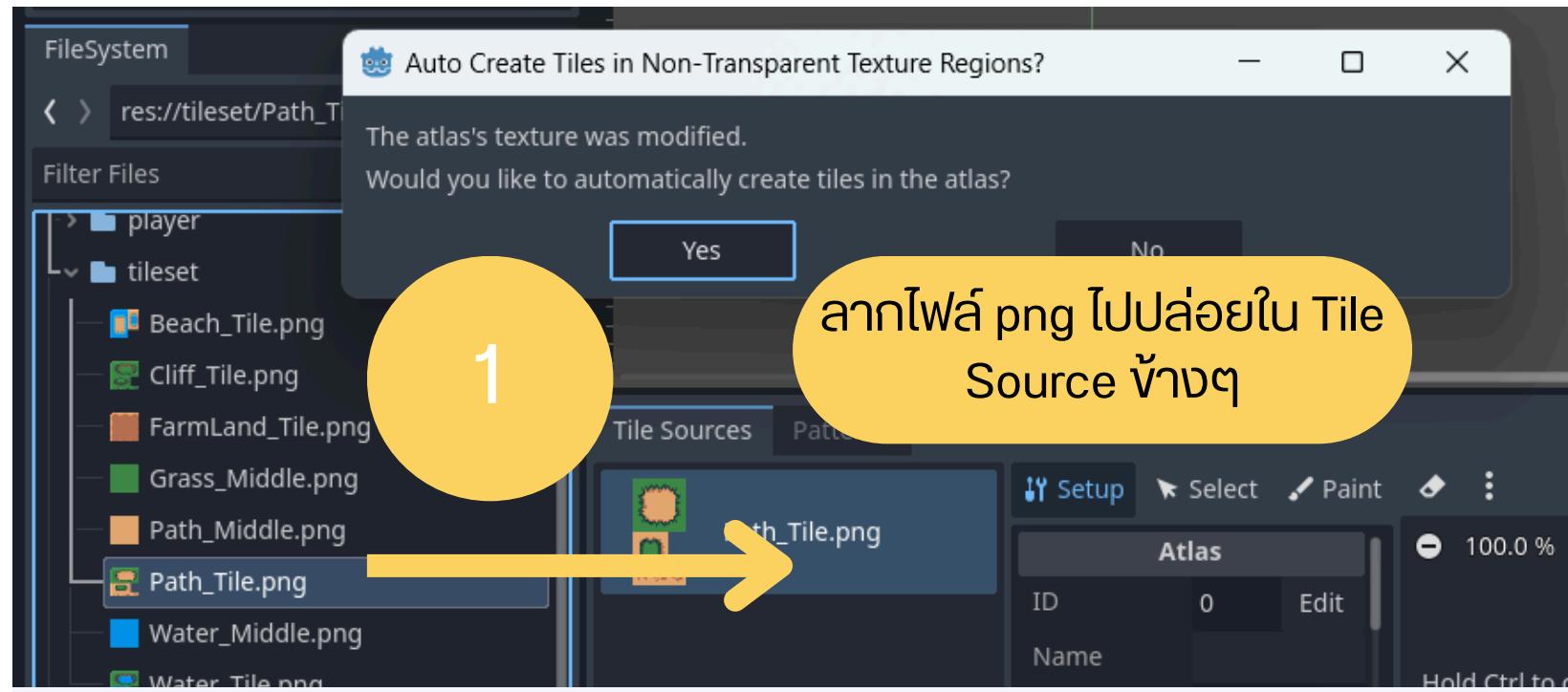
คลิกที่ TileMap ที่ Hierarchy ของ Panel Scene



ไปที่ Inspector คลิก DropDownList ของ Tile Set
เลือก Create **New Tileset**

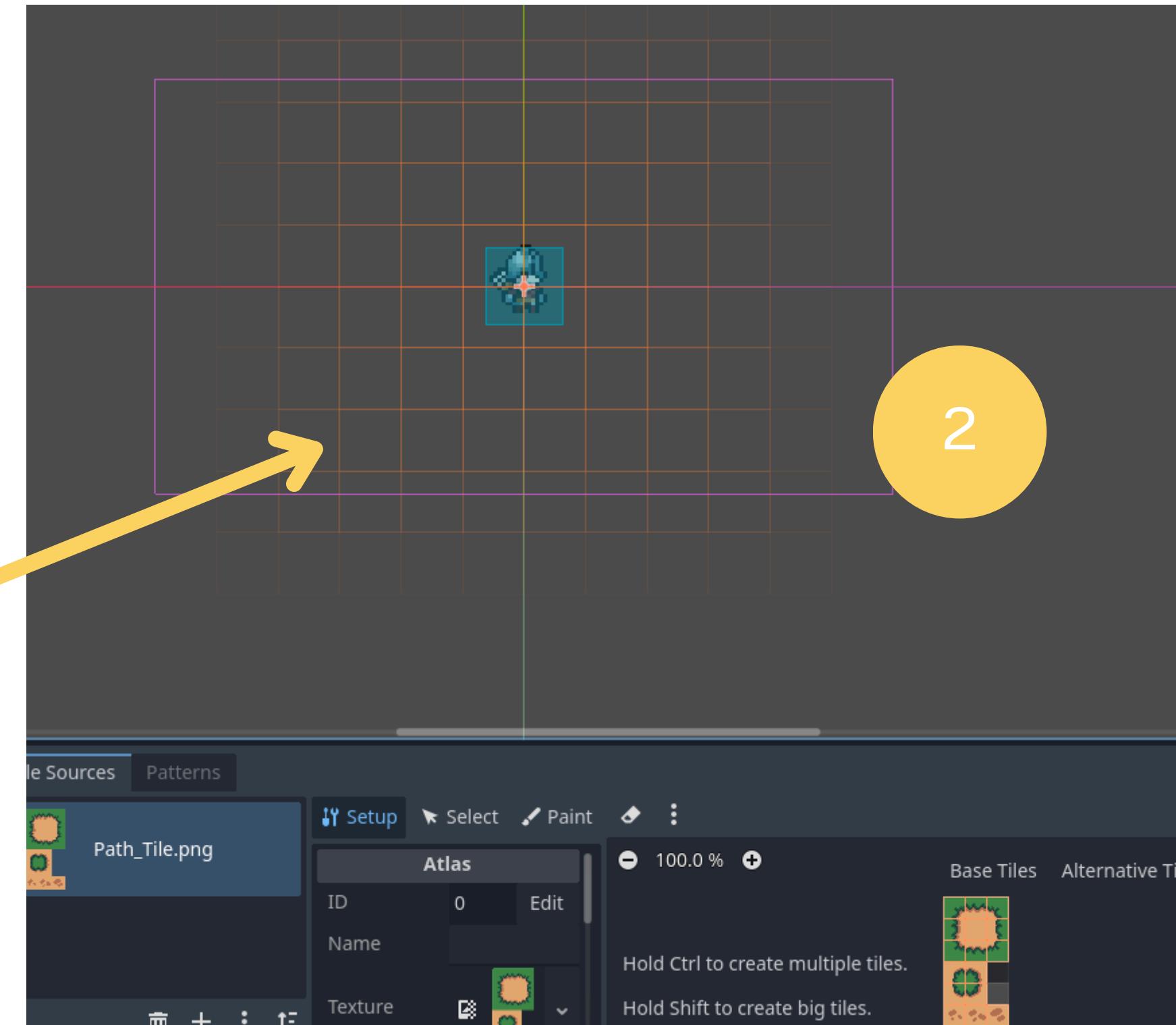
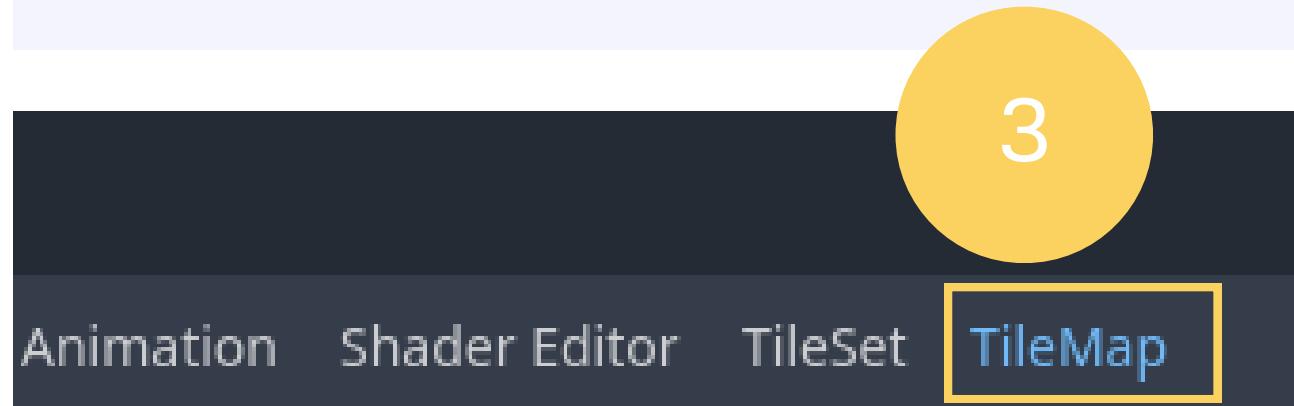


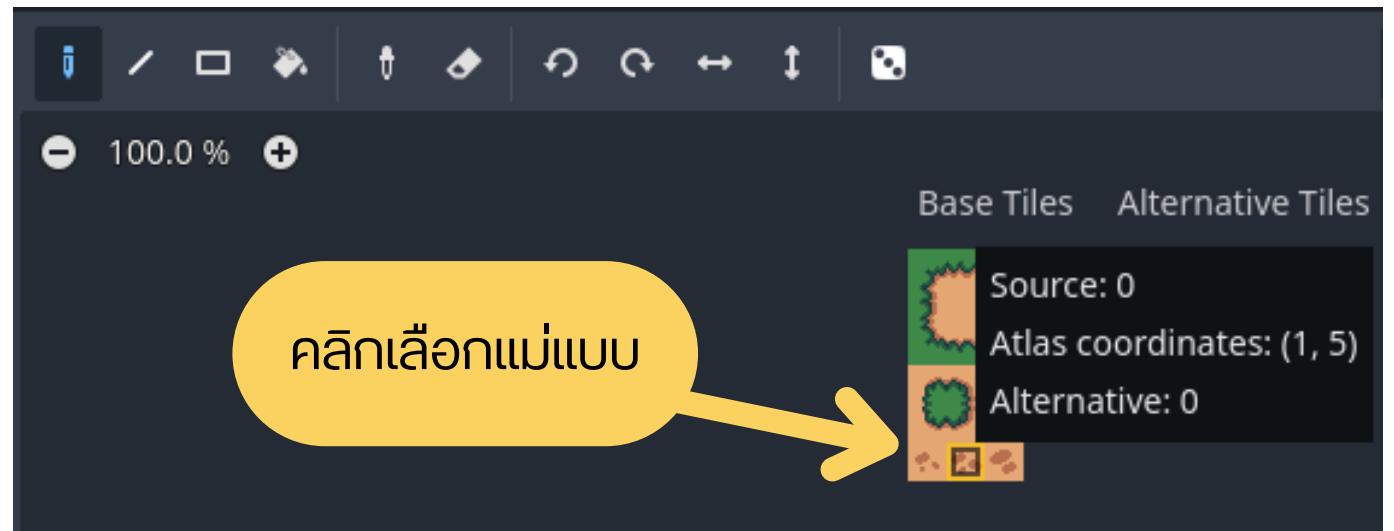
บุ Panel ของ Tile Sources จะถูกเพิ่มขึ้นมาให้เราไปทำงานที่แก็บกีซื้อ ว่า
TileSet ซึ่งต้องนำรูปภาพของเราใส่เข้าไป



ขั้นตอนที่ 2 การกำหนด TileSources

การใช้งานตัวแก้ไข TileSet Editor ของ Godot จะทำให้เราแบ่งภาพออกเป็น Tile ย่อย เช่น 16x16 หรือ 32x32 หรือ ขนาดเท่าๆ กันอยู่กับ Source รูปภาพของเราโดยมีหน่วยเป็นพิกเซล ระบบจะสร้าง Virtual Grid ให้ Editor ขึ้นมาให้เราสามารถดู และระบายน้ำได้อิสระ





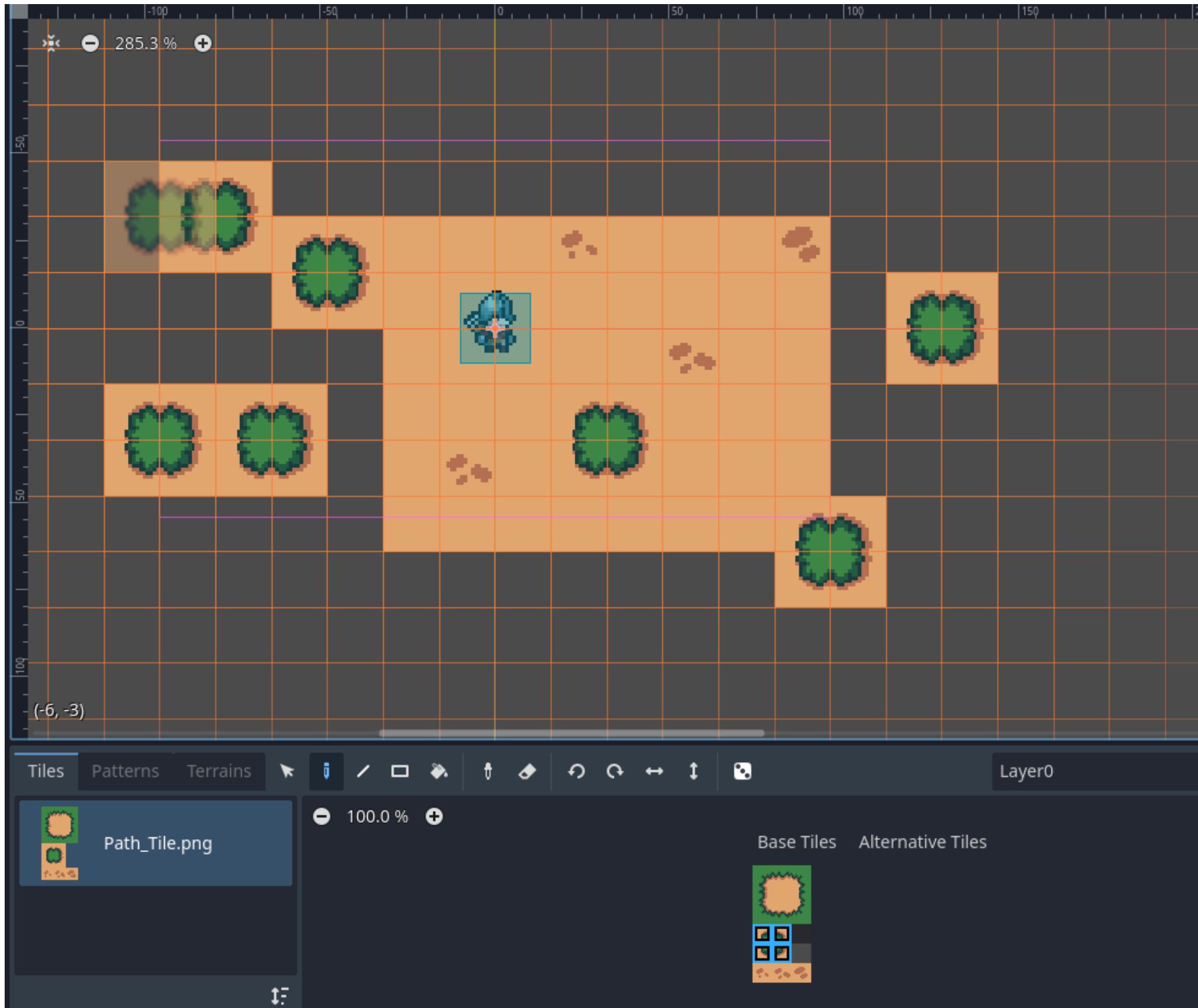
ขั้นตอนที่ 3 การ Paint Texture TileMap

การระบายแผนกีบบn Editor เราต้องเลือก แมปพิมพ์จากไฟล์ภาพ ของเราตามตัวแหนบ Box กีกำหนดหลังจากนั้นให้เลือกไอคอนเครื่องมือ ดินสอ สำหรับระบายสีของจากกีเรารอกแบบขึ้นมา

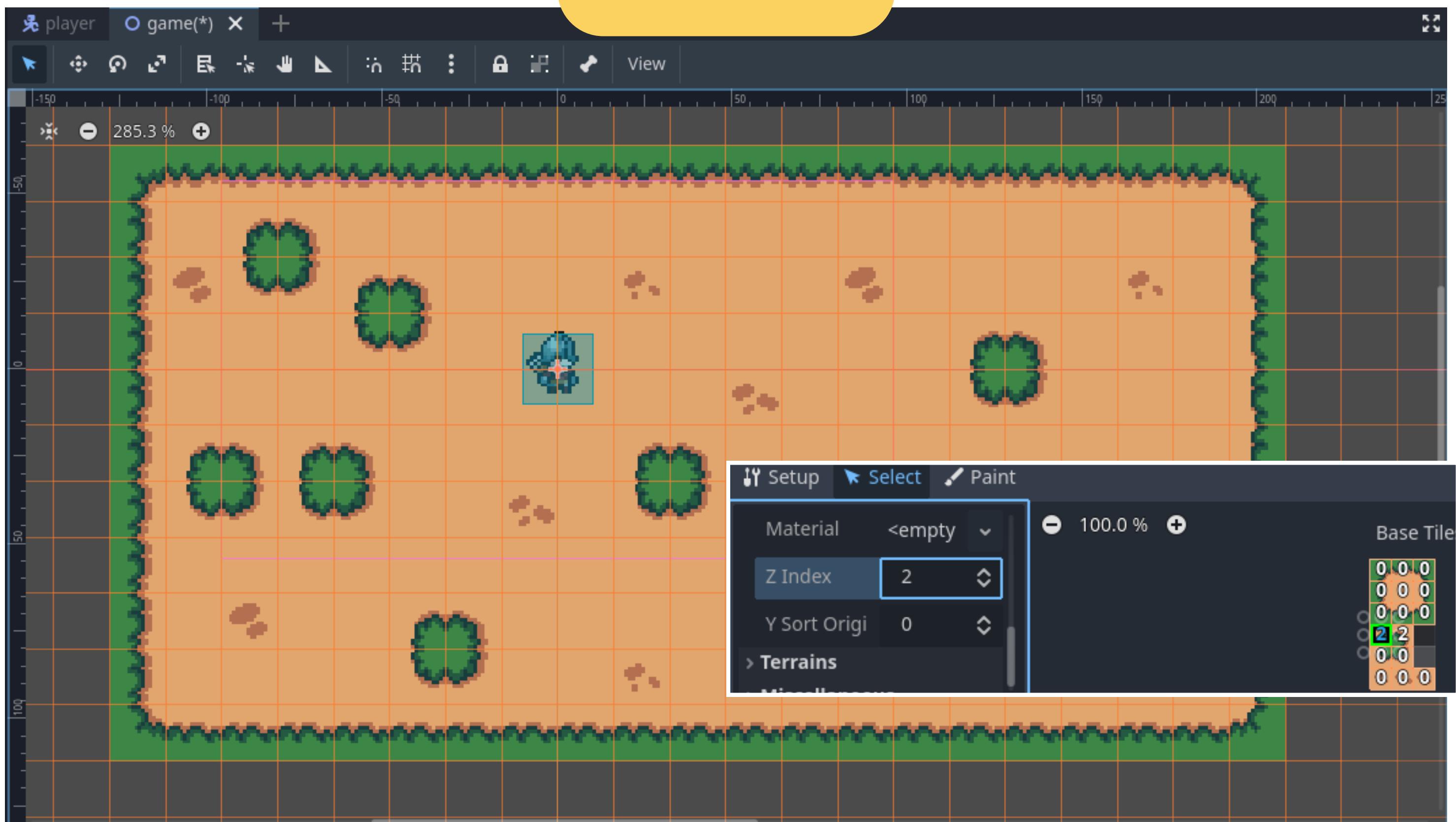


Challenge

ลองใช้เวลาออกแบบจากของเราจาก TileMap กีมี หรือจะดาวน์โหลดรูปภาพจาก tileSet ใหม่ๆ มาใช้งาน ก็ได้ จำกัดกีก็ว่างระดับหนึ่งเพื่อให้ตัวละครได้เดินไปมาได้บวก

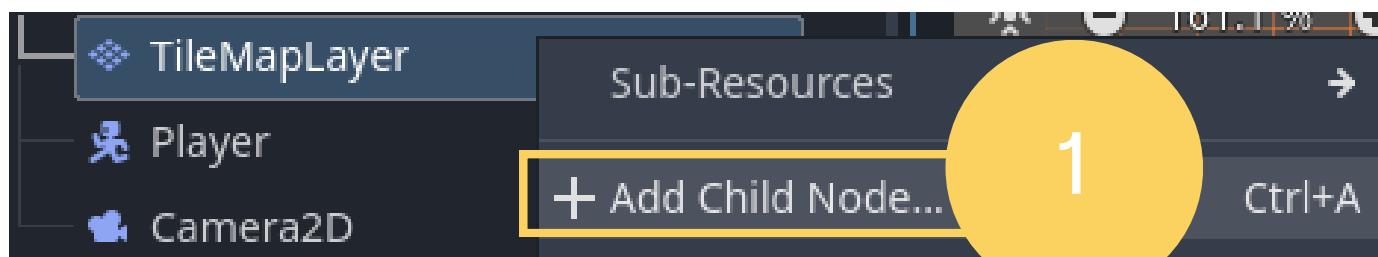


Outcome

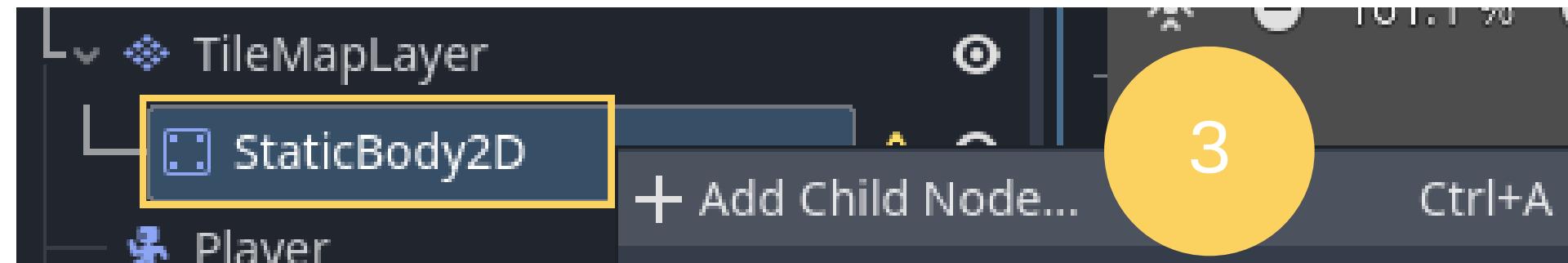
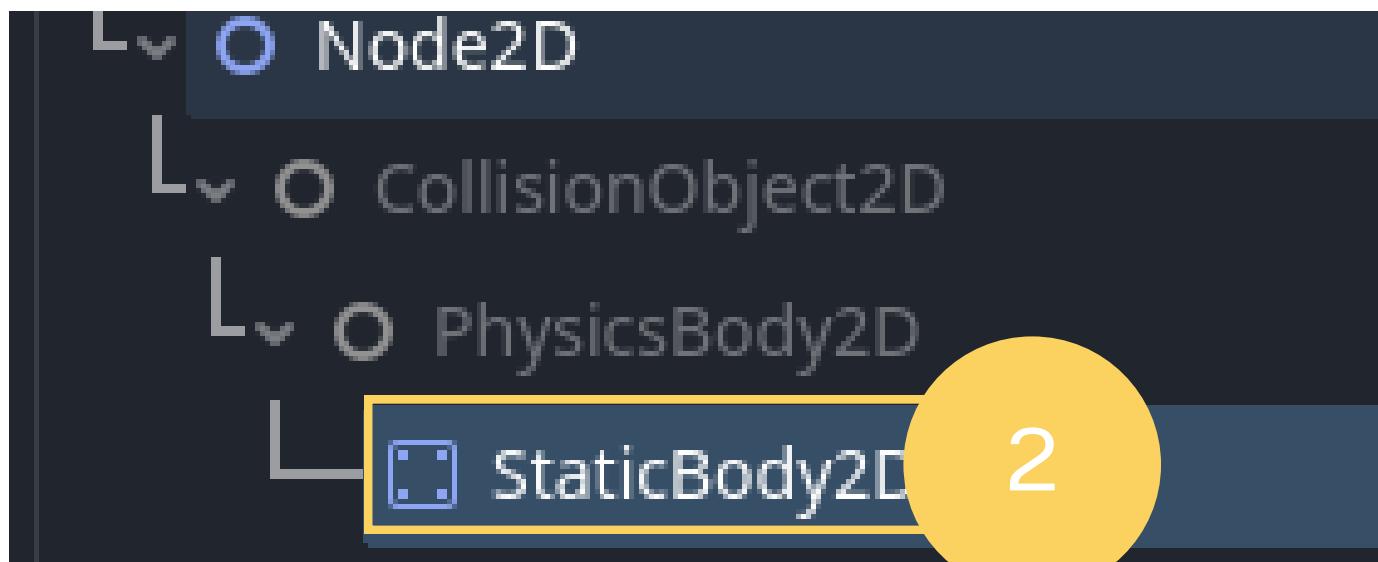


ขั้นตอนที่ 4 การกำหนด Boundary ของ Map

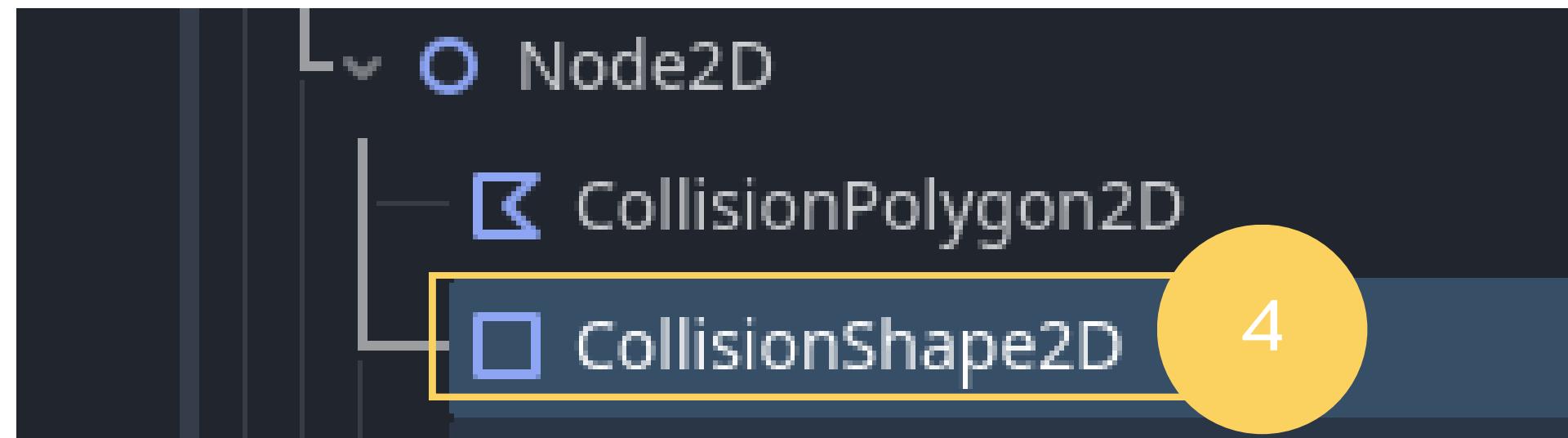
การกำหนด Boundary คือการกำหนดให้ตัวละครของเราเดินไปถึงจุดหนึ่ง แล้วไปต่อไม่ได้ เส้นบนมีกำแพงมากันไว้ เราสามารถทำได้เพื่อกำหนดขอบเขตการเคลื่อนที่ของตัวละครในเกมเราให้เดินไปตามเส้นทางได้ตามที่เรากำหนดมากกว่าปล่อยให้เดินเก็บจุดที่ต้องการ



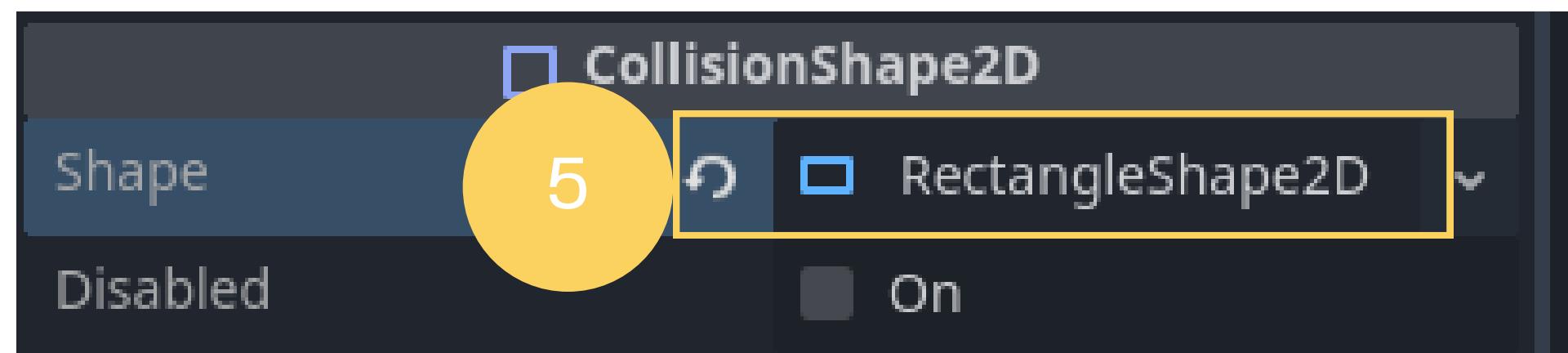
คลิกขวาที่ **TileMapLayer** เลือก **+ Add Child Node...** หลังจากนั้นเลือก **StaticBody2D** ขึ้นมา

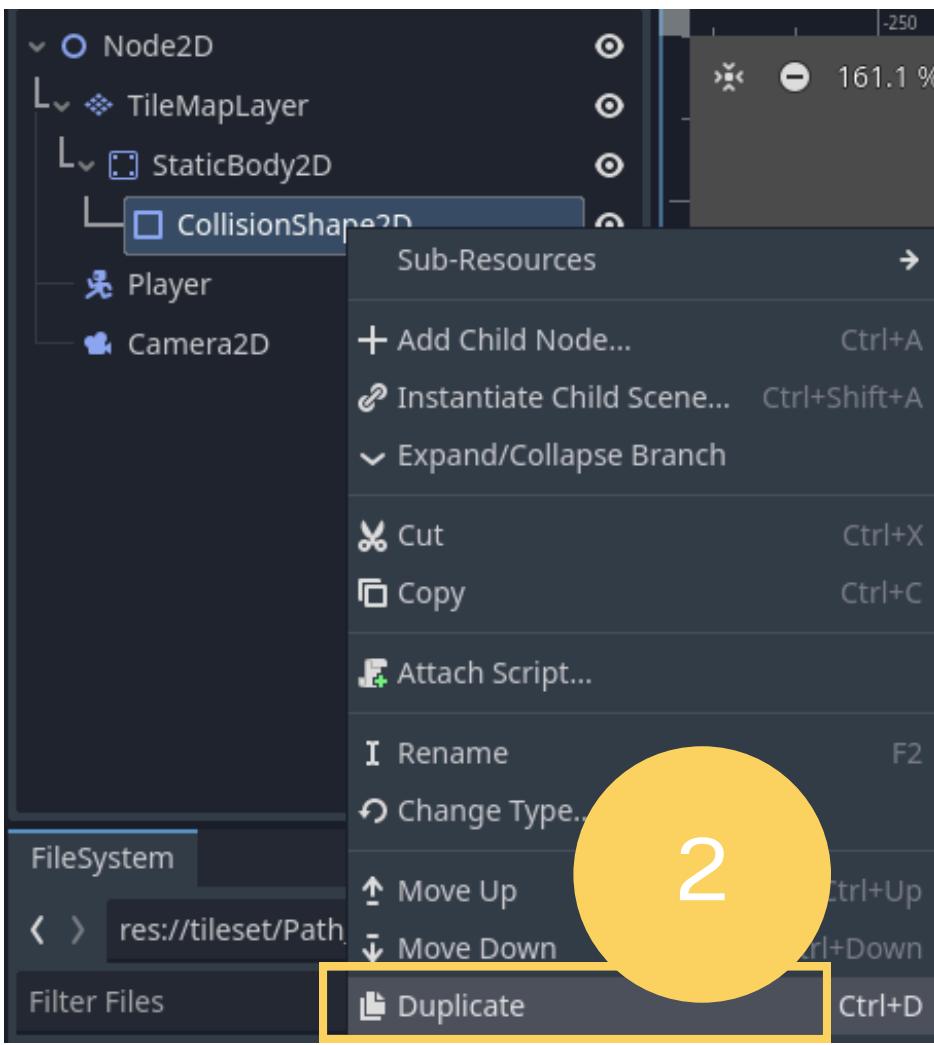
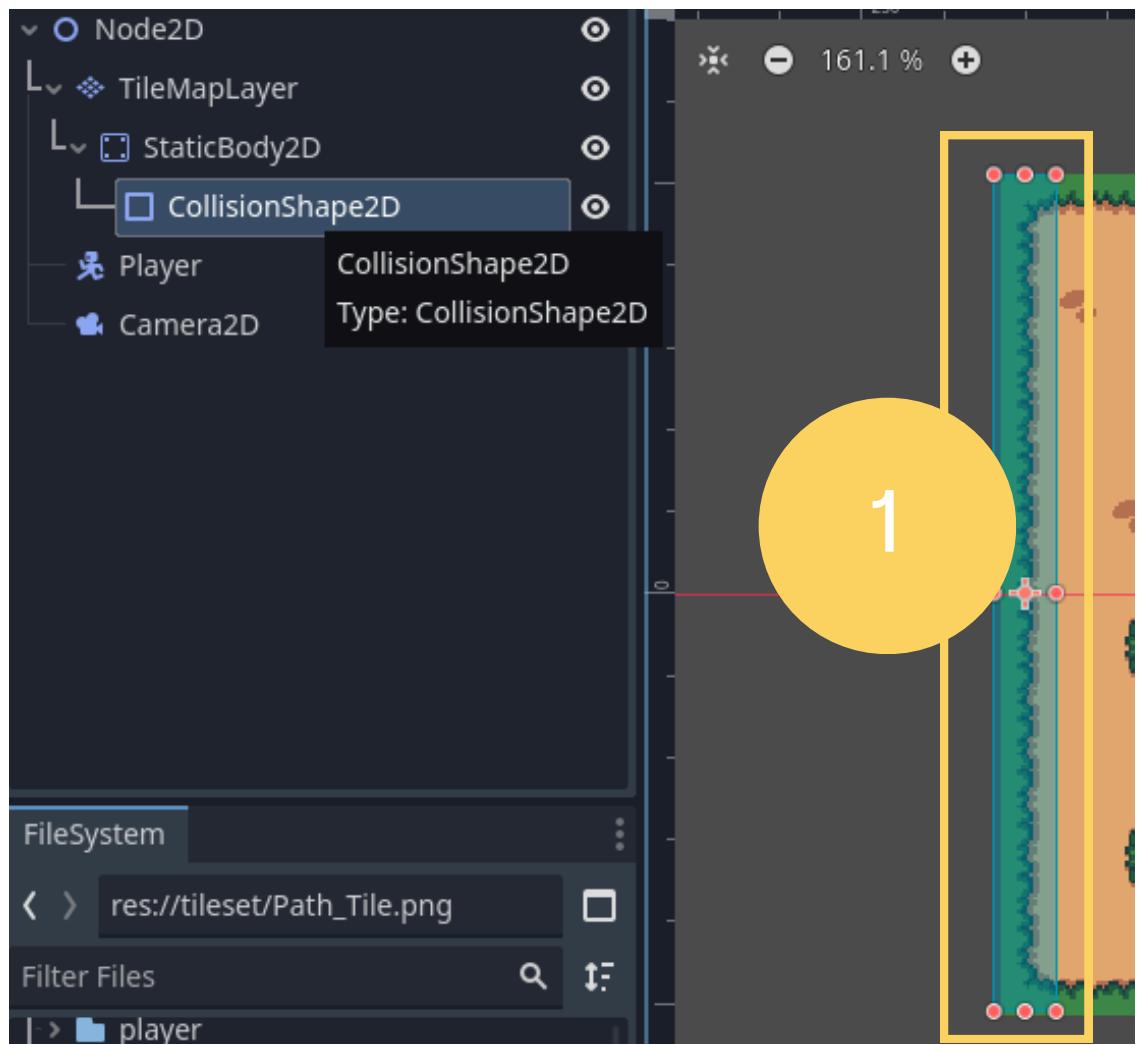


คลิกขวาที่ **StaticBody2D** เลือก **+ Add Child Node...** หลังจากนั้นเลือก **CollisionShape2D** ขึ้นมาเป็นลูกของมัน ลำดับขั้นตรวจสอบได้ที่รูปด้านล่าง

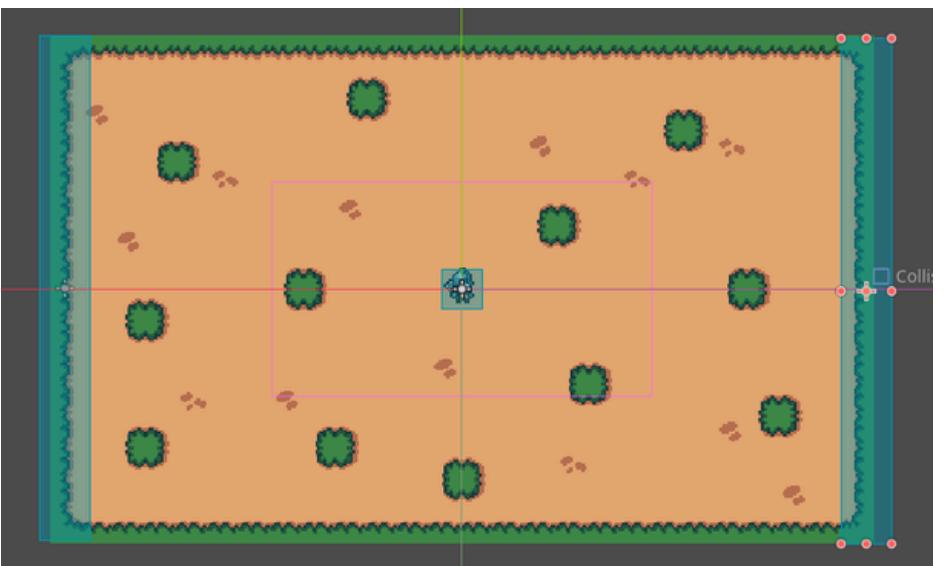


เลือก CollisionShape2D เป็นลูกของ StaticBody2D เลือก Shape เป็น Rectangle



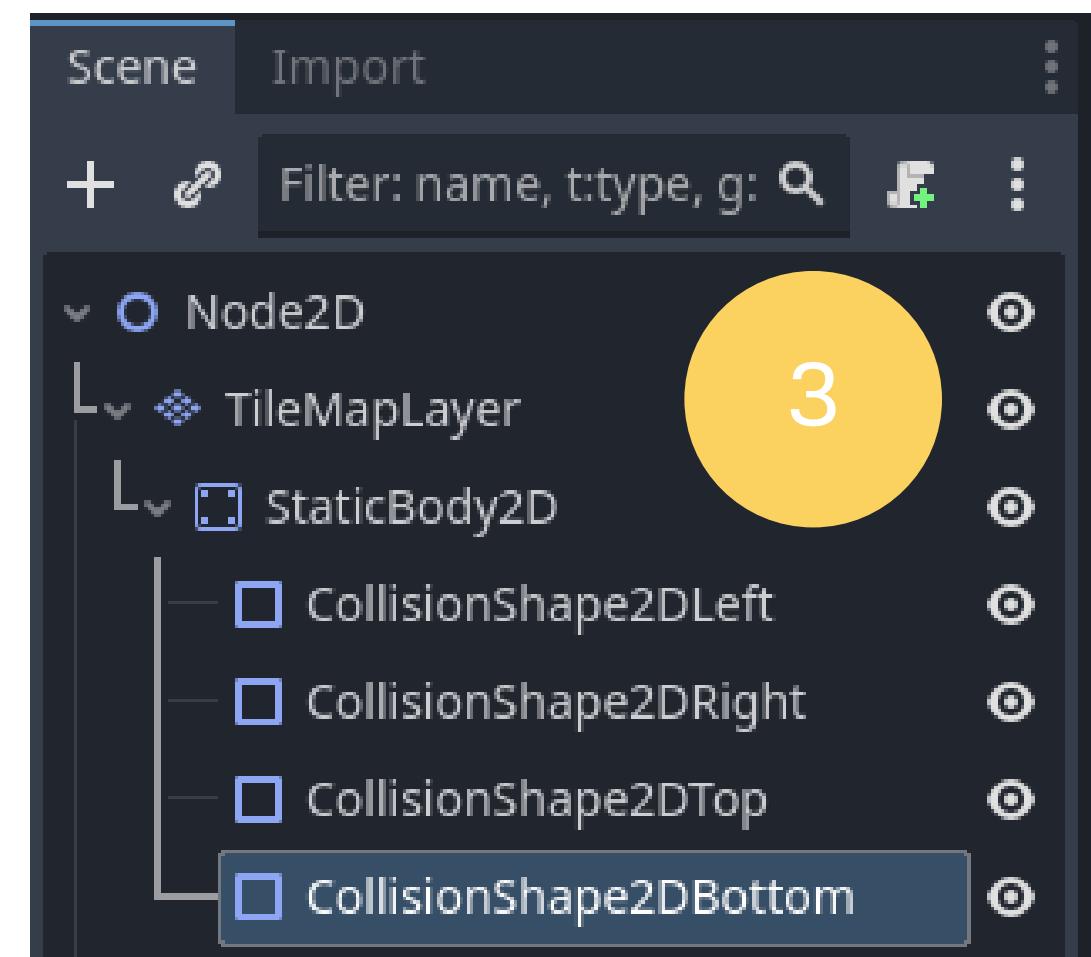
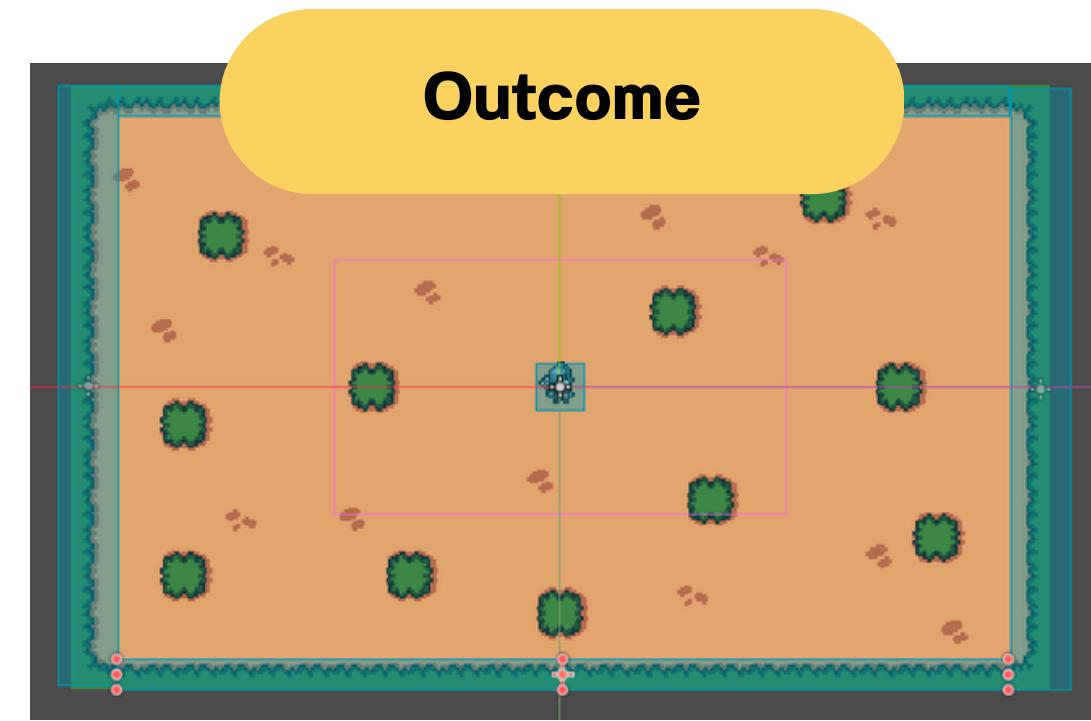


ปรับ Shape ของ Rectangle ให้พอดีกับด้านข้างของ Boundary ใน Scene ของเรา



Challenge

เพิ่ม CollisionShpae2D ด้านบน และด้านล่าง อีกรอบนึงด้วยตัวเอง กับกระบวนการทำงานที่กำลังมา



ขั้นตอนที่ 5 เขียน Code ควบคุมการแสดง AnimationPlayer ท่าเดิน ท่ายืน

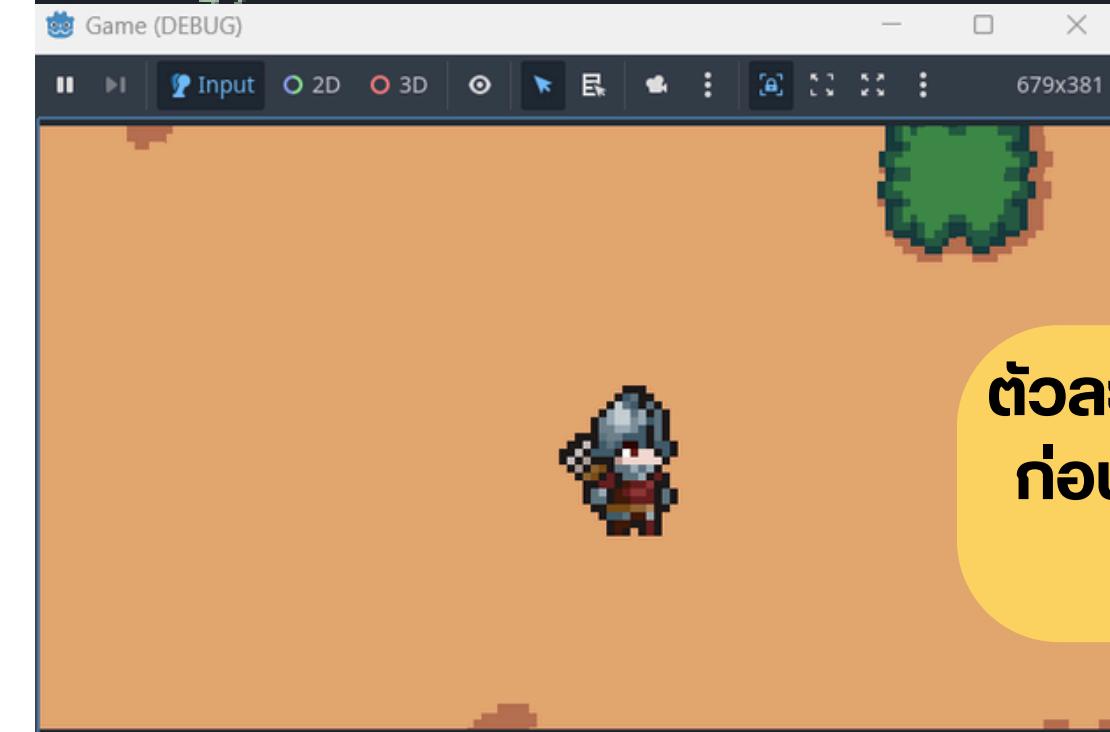
การเล่นแอปเปิเมชันโดยเบื้องต้นเราจะมี “Idle” และ “Walk” เป็นท่าพื้นฐานที่สร้างไว้แล้วเราจะนำมาเขียนโปรแกรมร่วมกัน เปิดไฟล์ playercontroller.gd ขึ้นมาแก้ไขเพิ่ม Code ดังนี้:

```
28    >  >
29    >  # เลือกแอปเปิเมชันให้เหมาะสมกับสถานะ
30  <  >  if direction == Vector2.ZERO:
31    >  >  $AnimationPlayer.play("Idle")
32  <  >  else:
33    >  >  $AnimationPlayer.play("Walk")
```

วางแผนของ Code

```
# เลือกแอปเปิเมชันให้เหมาะสมกับสถานะ
if direction == Vector2.ZERO:
    $AnimationPlayer.play("Idle")
else:
    $AnimationPlayer.play("Walk")
```

```
22
23    >  # สั่งให้ตัวละครเคลื่อนที่ตามพิกัด
24    >  move_and_slide()
25    >
26  <  >  if direction.x != 0:
27    >  >  $Sprite2D.flip_h = direction.x < 0
28    >  >
29    >  # เลือกแอปเปิเมชันให้เหมาะสมกับสถานะ
30  <  >  if direction == Vector2.ZERO:
31    >  >  $AnimationPlayer.play("Idle")
32  <  >  else:
33    >  >  $AnimationPlayer.play("Walk")
```

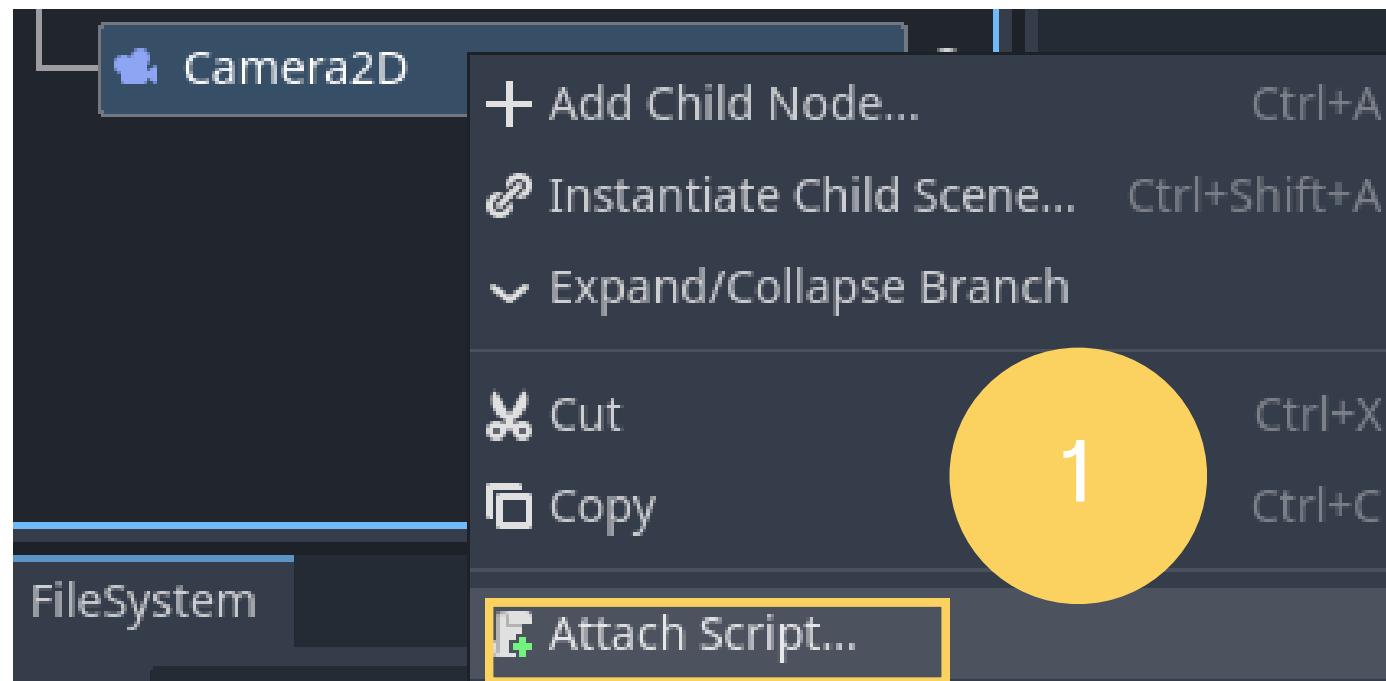


เอาไว้กวนใช้ tab ห้าม
เคาะ space

ตัวละครจะแสดงท่า Idle
ก่อนแต่เดินแล้วจะเป็น
Walk

ขั้นตอนที่ 6 สร้างระบบกล้องติดตามตัวละคร

กล้อง Camera2D ของเรารองมีการเคลื่อนที่ติดตามตัวละครอ้างอิง node เช่น “**Player**” บน Scene ของเรายังไห้คลิก Camera2D บน **game.tscn** หลังจากนั้น Attach Script ตั้งเช่น **cameracontroller.gd** ขึ้นมา



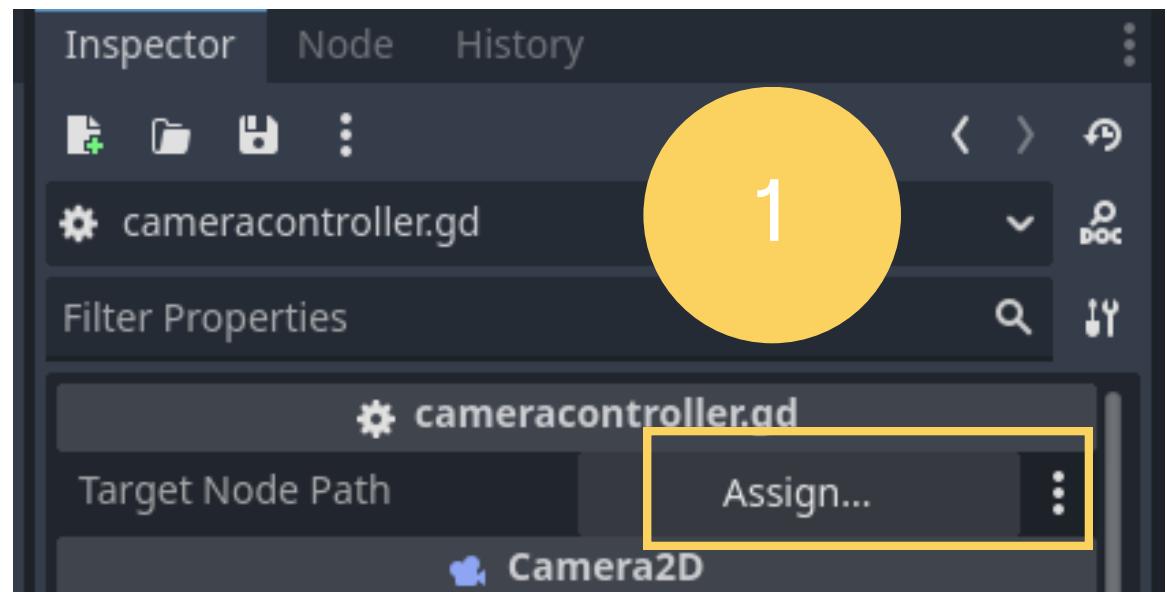
```
1  extends Camera2D
2
3  @export var target_node_path: NodePath = NodePath("")
4
5  var target: Node2D
6
7  func _ready() -> void:
8    if target_node_path != NodePath(""):
9      target = get_node_or_null(target_node_path)
10
11 func _process(delta: float) -> void:
12  if target:
13    global_position = target.global_position
```

extends Camera2D

@export var target_node_path: NodePath = NodePath("")
var target: Node2D

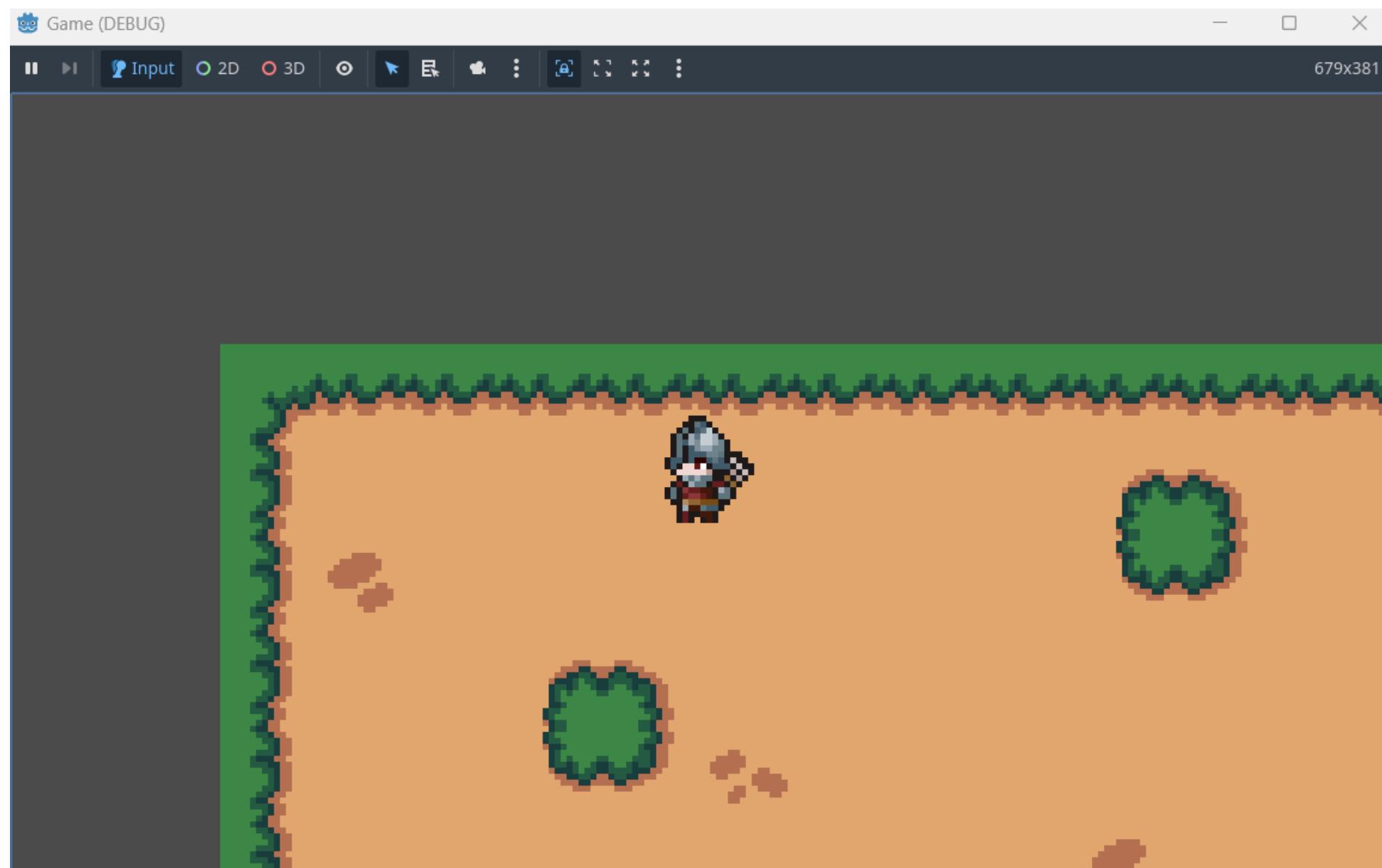
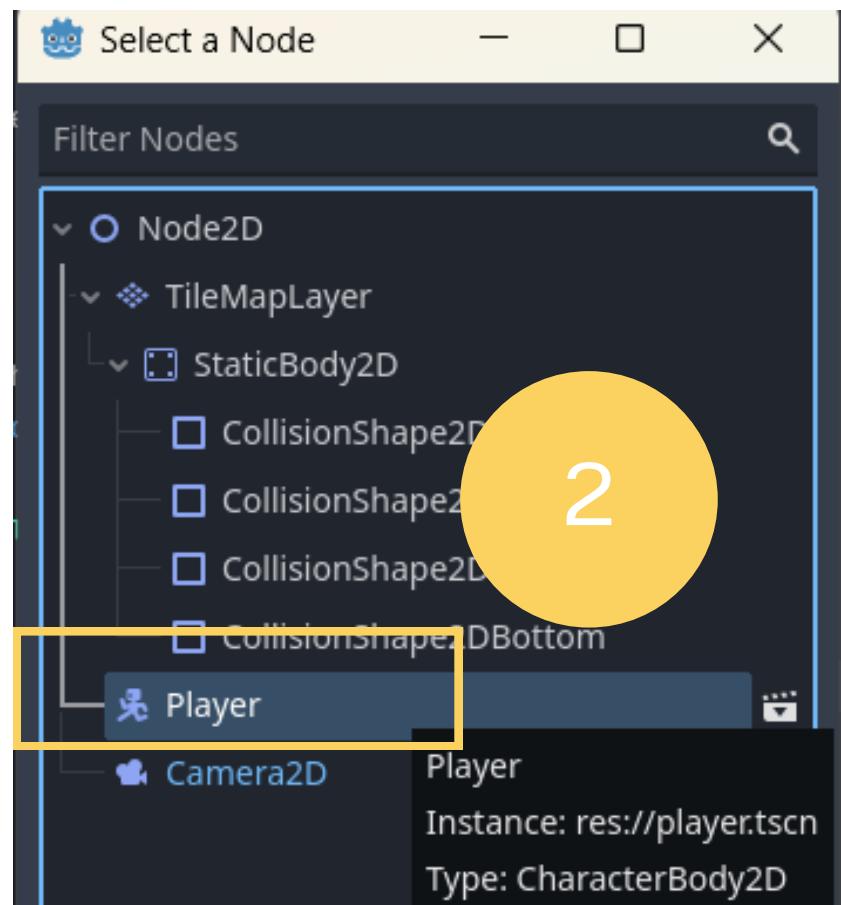
func _ready() -> void:
if target_node_path != NodePath("":
target = get_node_or_null(taget_node_path)

func _process(delta: float) -> void:
if target:
global_position = target.global_position



เลือก Camera2D ใน Inspector

ໃນช่อง Target Node Path → คลิกเลือกไปยัง
Node ตัวละคร (เช่น Player)



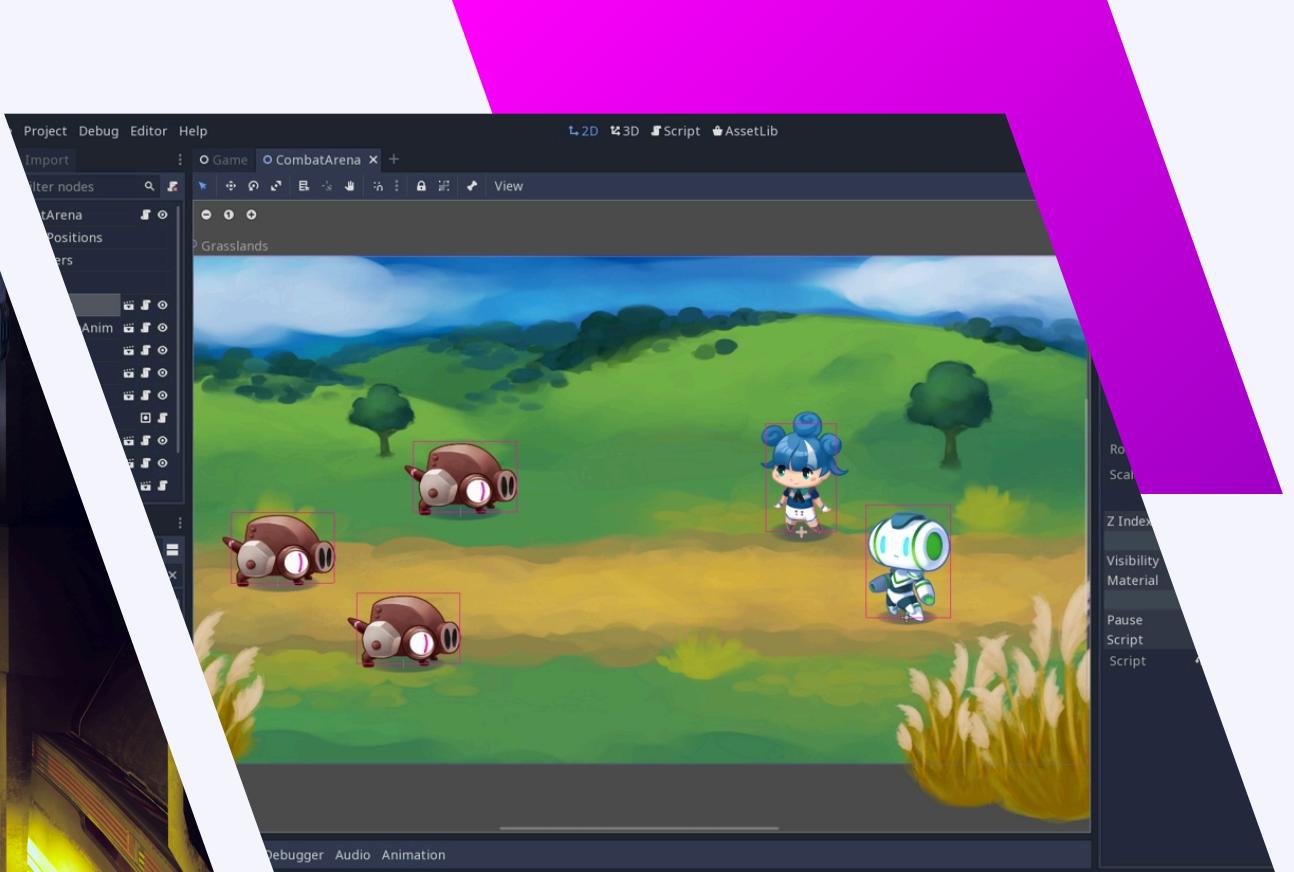
ทดสอบระบบของเราว่าเห็นว่าตอนนี้ ตัวละครของเรา มีกล้องเคลื่อนที่ติดตามเราแล้ว

Source Code สามารถดาวน์โหลดตรวจสอบได้ที่:

<https://github.com/banyapon/antdpugodotcourse/tree/main/chapter2>

GAME DEVELOPER

College of Creative Design and Entertainment Technology



การพัฒนาเกมด้วยโปรแกรม Godot Engine