

หลักสูตร: **Unity 6** สร้างเกม **2D**
Platformer ตั้งแต่เริ่มต้น

บทที่ 1: ทำความรู้จัก Unity 6 และเตรียมโปรเจกต์

บทที่ 2: ออกแบบฉากเกม 2D ด้วย Tilemap

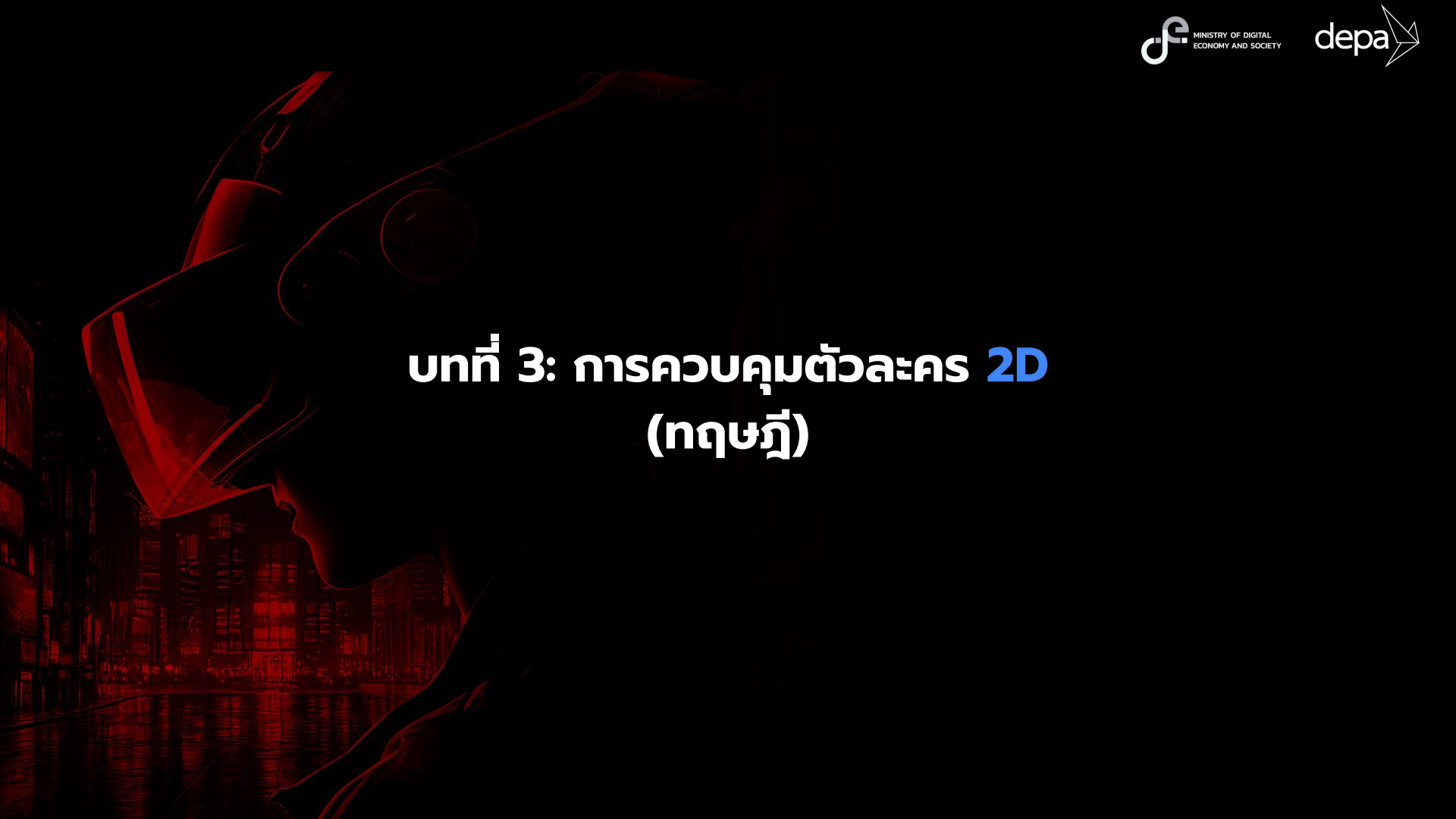
► **บทที่ 3: การควบคุมตัวละคร 2D**

บทที่ 4: ระบบเก็บเหรียญ และศัตรู

บทที่ 5: ระบบการต่อสู้

บทที่ 6: ระบบ UI และ HUD

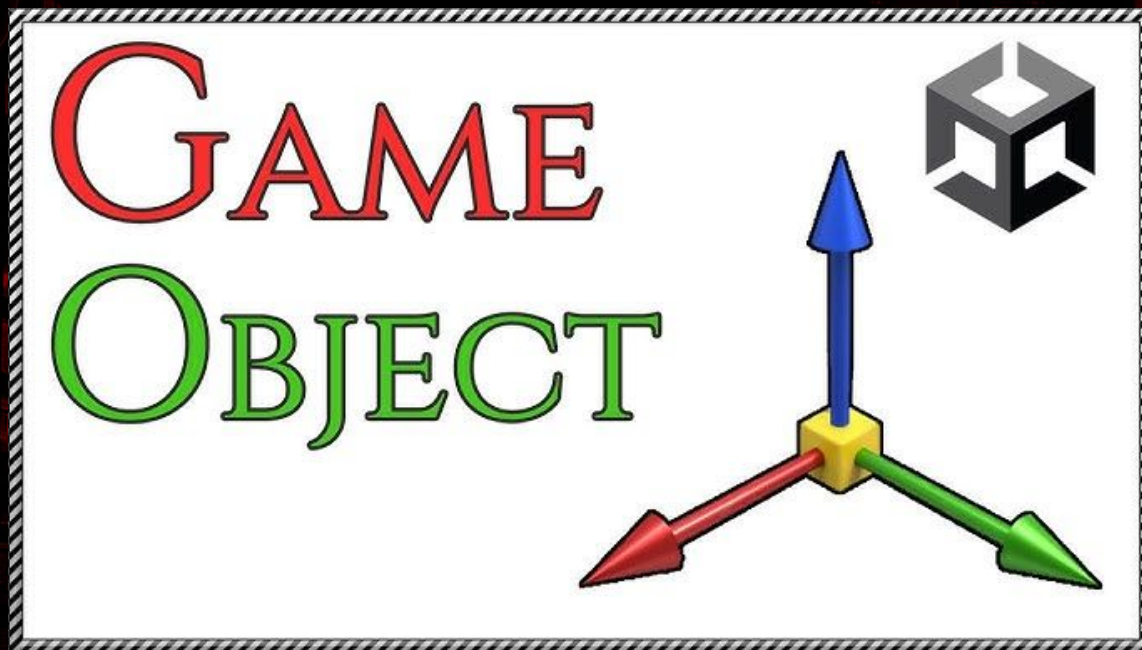
บทที่ 7: Build เกมออกมาให้เล่นจริง



บทที่ 3: การควบคุมตัวละคร 2D (ทฤษฎี)

GameObject คืออะไร

GameObject หมายถึง "วัตถุพื้นฐาน" (Basic Object) ที่ใช้สร้างทุกสิ่งในเกม ไม่ว่าจะเป็นตัวละคร ศัตรู ไอเท็ม ฉาก หรือแม้แต่กล้องและแสงไฟ



คุณสมบัติหลักของ **GameObject**

เป็นกล่องเปล่า

- **GameObject** เริ่มต้นเป็นเพียง "คอนเทนเนอร์" ว่าง ๆ ที่ไม่มีรูปร่างหรือพฤติกรรม
- การจะทำให้มัน "มีประโยชน์" ต้องเพิ่ม **Component** เข้าไป

ทำงานผ่าน **Component**

- ตัวอย่าง **Component** ที่มักใช้:
 - **Transform** (ทุก **GameObject** ต้องมีเสมอ): กำหนดตำแหน่ง (**Position**), การหมุน (**Rotation**), และขนาด (**Scale**)
 - **Sprite Renderer**: ทำให้ **GameObject** แสดงผลเป็นภาพ 2D
 - **Rigidbody2D**: ทำให้ **GameObject** ถูกควบคุมด้วยฟิสิกส์ เช่น แรงโน้มถ่วงหรือการชน
 - **Collider2D**: ใช้ตรวจสอบการชนกับวัตถุอื่น
 - **Script (C#)**: ใส่พฤติกรรมเฉพาะที่เรากำหนดเอง

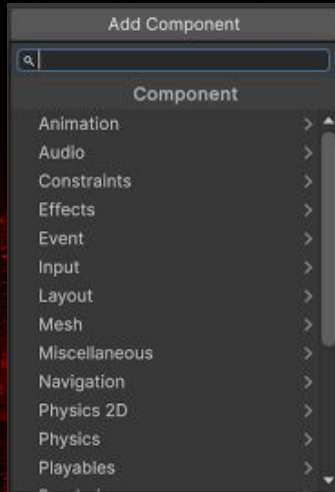
คุณสมบัติหลักของ **GameObject**

ยืดหยุ่นมาก

- **GameObject** หนึ่งสามารถมี **Component** ได้หลายตัว เพื่อรวมคุณสมบัติต่าง ๆ เข้าด้วยกัน
- เช่น “ตัวละครผู้เล่น” อาจมี:
 - **Sprite Renderer** → แสดงภาพตัวละคร
 - **Rigidbody2D** → เคลื่อนที่ตามฟิสิกส์
 - **Collider2D** → ตรวจสอบการชน
 - **PlayerController.cs** → ควบคุมการเดิน/กระโดด

เปรียบเทียบให้เข้าใจง่าย

- **GameObject** = หุ่นยนต์เปล่า
- **Component** = ชิ้นส่วน / โมดูล ที่เราติดตั้งเพิ่ม เช่น กล้อง, แขน, ล้อ, ระบบควบคุม
- ถ้าไม่มี **Component** → **GameObject** จะไม่ทำอะไรเลย
- ถ้ามี **Component** → **GameObject** จะเริ่มมี "หน้าที่" ตามที่ใส่เข้าไป



ควบคุมการเคลื่อนไหว (Character Movement)

คือหัวใจสำคัญที่ทำให้ตัวละครผู้เล่นสามารถ “เดิน กระโดด หรือเคลื่อนไหว” ได้ตามที่เรากดปุ่ม ไม่ใช่แค่เป็นภาพนิ่งอยู่ในฉาก

ทำไมต้องใช้สคริปต์

- **Unity** มีระบบฟิสิกส์และ **Input** ให้แล้ว แต่ตัวละครจะไม่รู้ว่าต้อง “เดินซ้าย-ขวา” หรือ “กระโดดเมื่อกด **Space**” ถ้าเราไม่เขียนโค้ดบอกมัน
- สคริปต์จึงทำหน้าที่เป็น “สมอง” ที่รับค่าจากคีย์บอร์ด/จอย → แปลงเป็นการเคลื่อนไหว

หลักการทำงาน

- รับ **Input** จากผู้เล่น (เช่น กดปุ่มลูกศรซ้าย/ขวา หรือปุ่ม **Space**)
- นำค่าที่ได้ไปเปลี่ยนความเร็ว (**Velocity**) ของตัวละครผ่าน **Rigidbody2D**
- ตรวจสอบเงื่อนไขพิเศษ เช่น อยู่บนพื้นแล้วถึงจะกระโดดได้

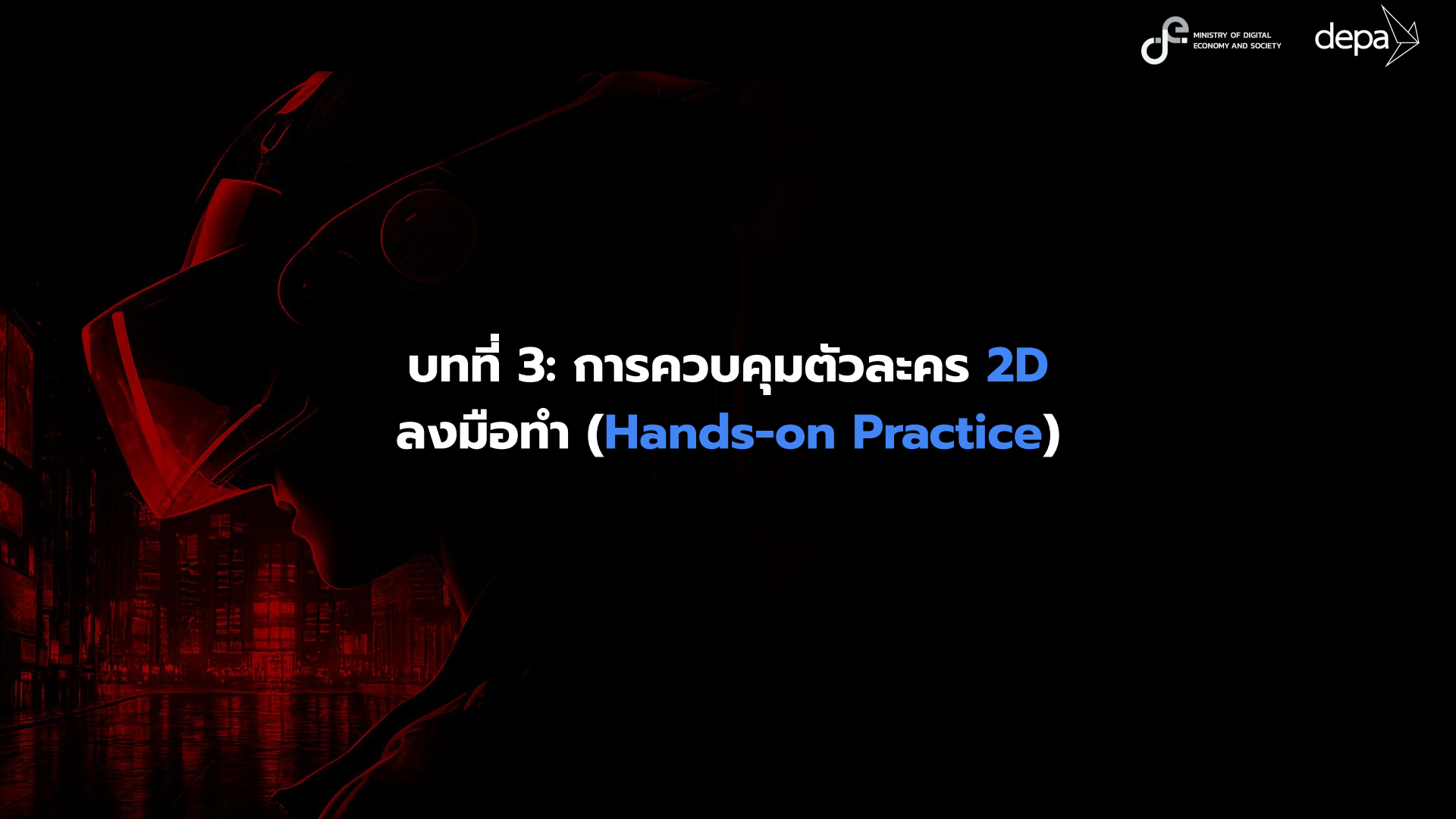
ควบคุมการเคลื่อนไหว (Character Movement Script)

องค์ประกอบหลักในโค้ด

- ตัวแปรควบคุมความเร็ว → ให้ตัวละครเคลื่อนที่ไม่เร็ว/ช้าเกินไป
- การตรวจสอบพื้น → ป้องกันกระโดดรัวกลางอากาศ
- การทำ **Flip** → หันตัวละครตามทิศที่เดิน
- **Animation Trigger** → เปลี่ยน **Animation (Idle, Run, Jump)**

2D PLAYER MOVEMENT





บทที่ 3: การควบคุมตัวละคร 2D ลงมือทำ (Hands-on Practice)