

www.gamedevpbru.com

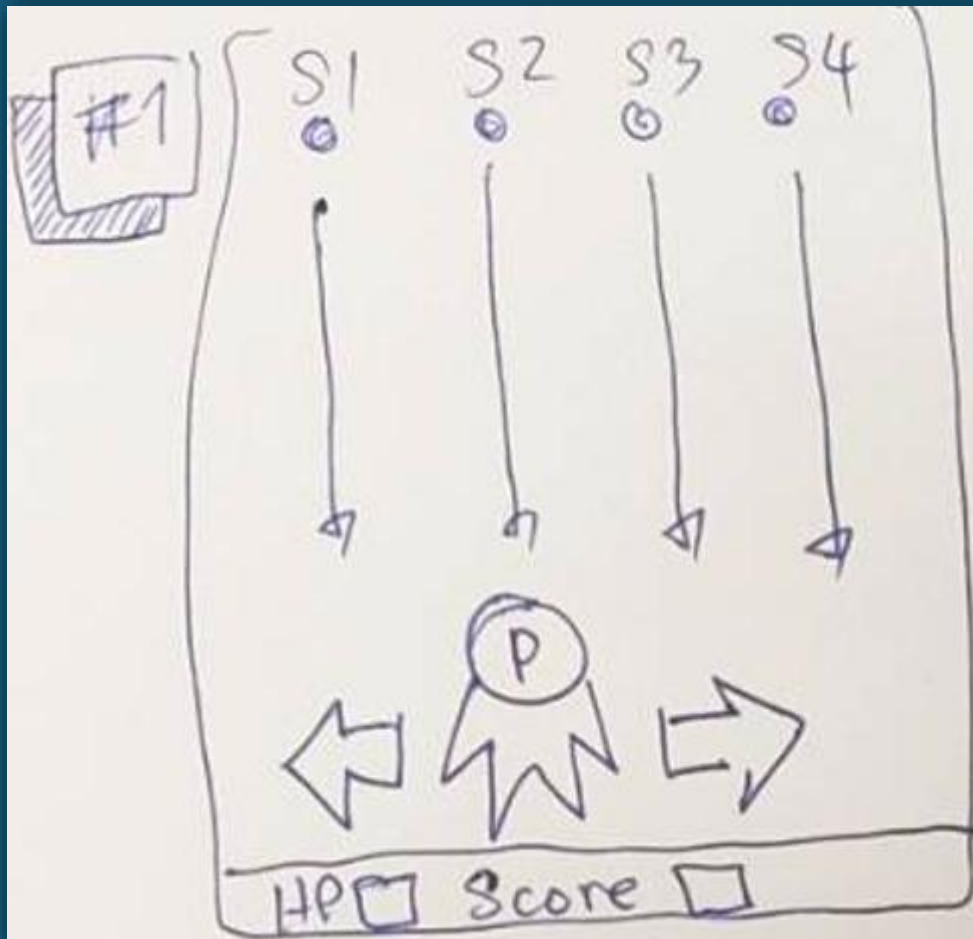
miniGameNo.1

ดนัย เวชฎาฐิติกุล

จารุต บุศราทิว

DcG.IT@PBRU





P = Player \rightarrow Controller

S1, S2, S3, S4 = \rightarrow $\begin{cases} \text{အကူပေး} & (+1) \\ \text{အခက်ခဲ} & (-1) \end{cases}$

Player Properties $\begin{cases} \text{HP} \\ \text{Max HP} \\ \text{Score} \end{cases}$





Properties

```
int hp;  
int hpMax;  
int score
```

Activities

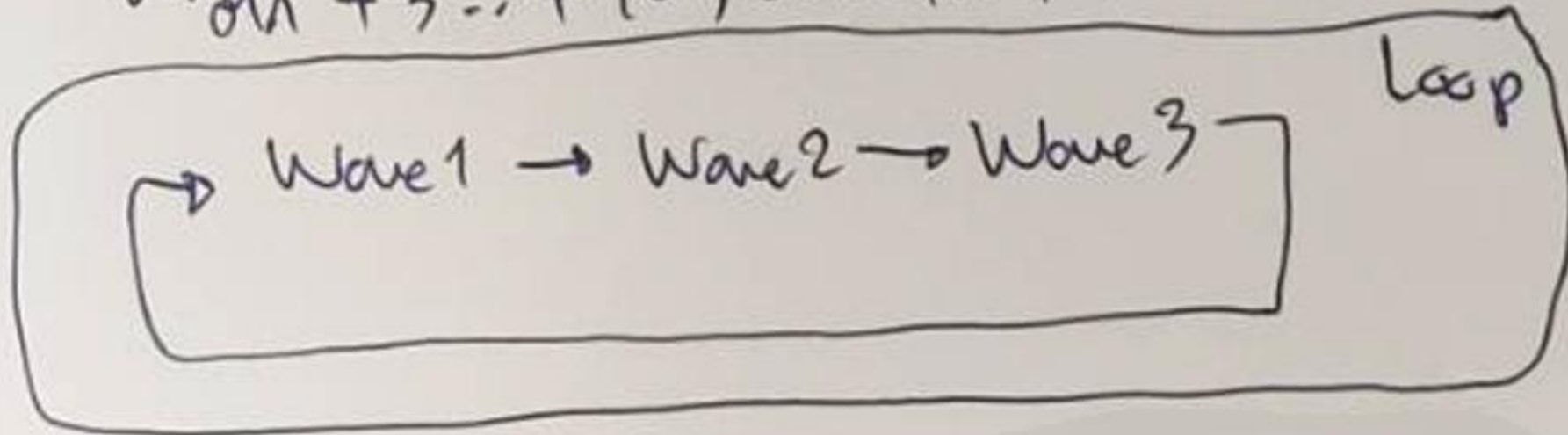
```
moveLeft() → ...  
moveRight() → ...  
incScore() → ...  
decScore() → ...  
update() → ...
```



Wave 1 → min 0-10
→ $0 \times 1 + 2$, $0 \times 2 + 1$ 1-2

Wave 2 → min 11-20
→ $0 \times 1 + 3 + 4$, $0 \times 2 + 1$ 1

Wave 3 → min 21-30
→ $0 \times 1 + 3 \dots + 10$, $0 \times 2 + 1$ 1



ଉଦାହରଣ

Score ୯୦% → ୧୦୦ HP + 1

Score ୮.10 → ୧୦୦ HP + 1



HP = 0

ଉତ୍ତର



เตรียมความพร้อม



ตรวจสอบ

- Unity 6.3.x
- สร้างโครงงานแบบ 3D
 - กำหนดให้บันทึกลงเครื่อง
 - ตั้งชื่อโครงงานเป็น miniGameNo1
- สร้างฉาก (Scene) ใหม่ ชื่อ miniGameNo1Scene





Unity 6.3 LTS (6000.3.3f1) LTS

▼ In progress (0 of 1 completed)



Editor application
Installing...

Installs

All

Official releases

Pre-releases

UNITY 6



Unity 6.3 LTS (6000.3.3f1) LTS

C:\Program Files\Unity\Hub\Editor\6000.3.3f1\Editor\Unity.exe

Windows



← New project

Templates

Select an Editor version, then choose a template to start your project.

Editor version *

6000.3.3f1 LTS

All

Core

Sample

Le: >

Search

Core



Universal 2D

Core

SRP

Universal 3D

Universal 3D

This template includes the settings and assets you need to start creating with the Universal Render Pipeline (URP).

[Read more](#)

Unity organization

sarmone3rd

Project name *

miniGameNo1

Location *

C:\Users\cid\Desktop\UnityImgSnd

Use Unity Version Control ?





SƏ

Unity Engine

6000.3.3f1

© 2025 Unity Technologies. All rights reserved.
Artwork made with Unity by Keijiro Takahashi.

miniGameNo1

Open Project: Load Scripting Assemblies



Hierarchy

- SampleScene
 - Main Camera
 - Directional Light
 - Global Volume

Scene Game

Center Global 1

Inspector

URP Empty Template

Welcome to the Universal Render Pipeline

This template includes the settings and assets you need to start creating with the Universal Render Pipeline.

URP Documentation

[Read more about URP](#)

Forums

[Get answers and support](#)

Report bugs

[Submit a report](#)

Remove Readme Assets

Asset Labels

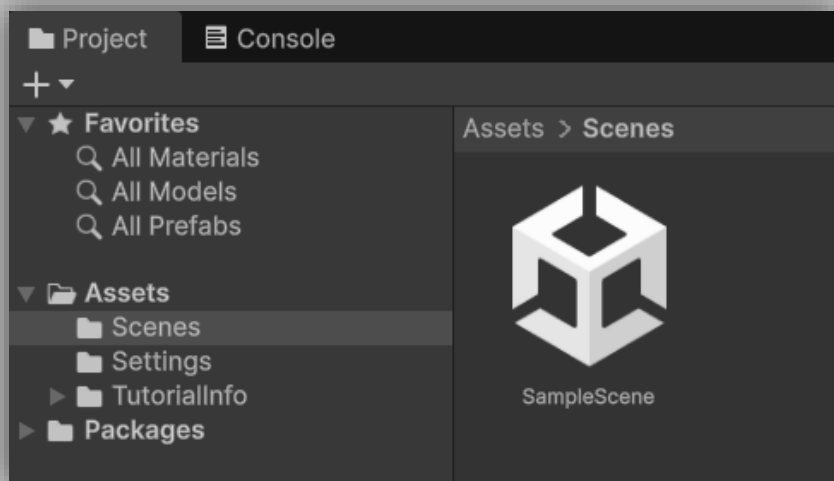
AssetBundle None None

Project Console

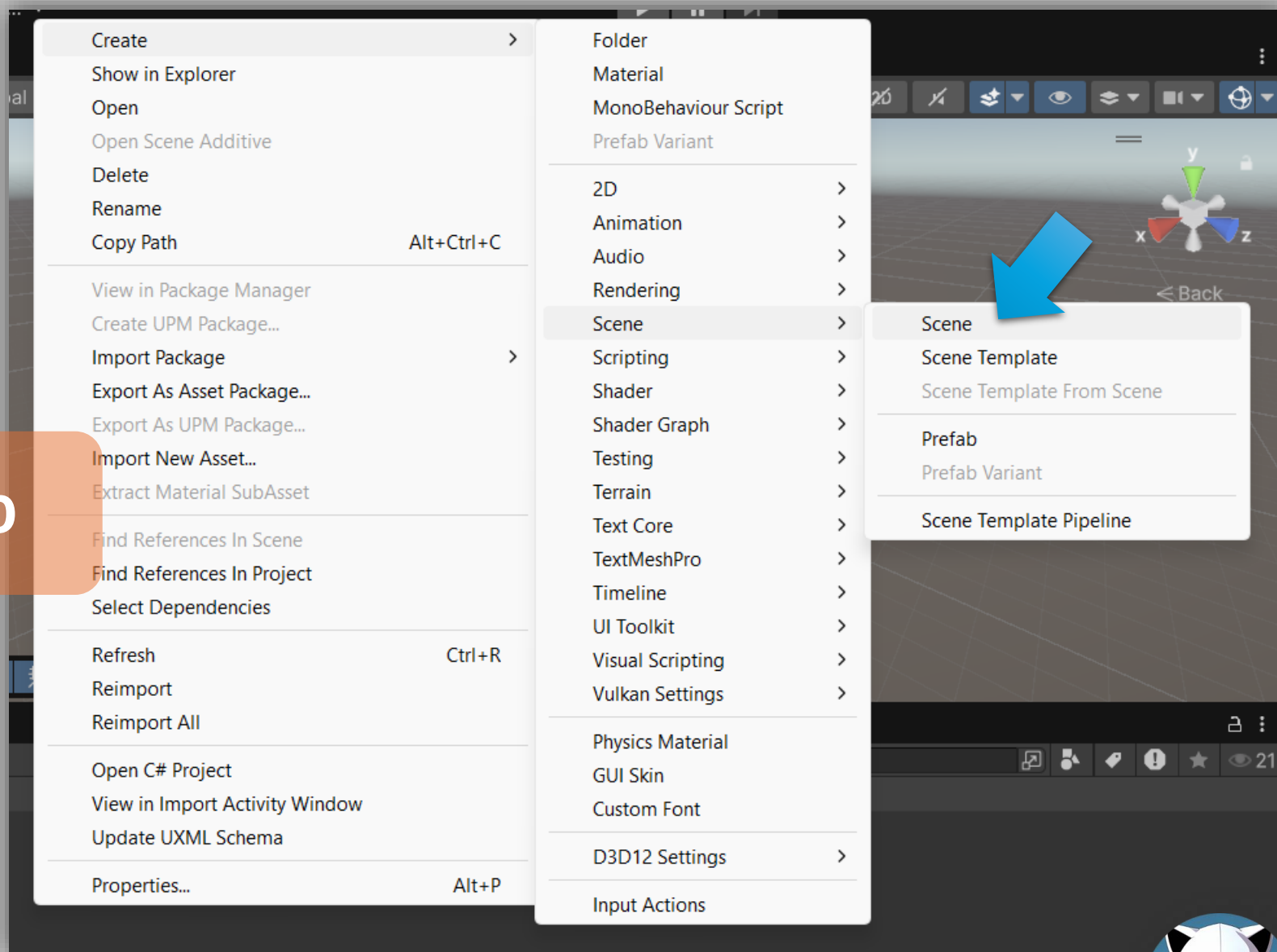
Clear Collapse Error Pause Editor

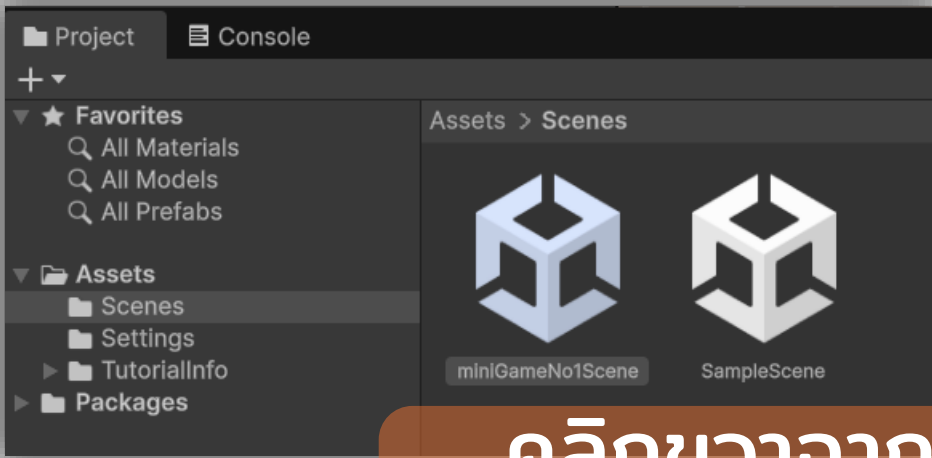
0 0 0



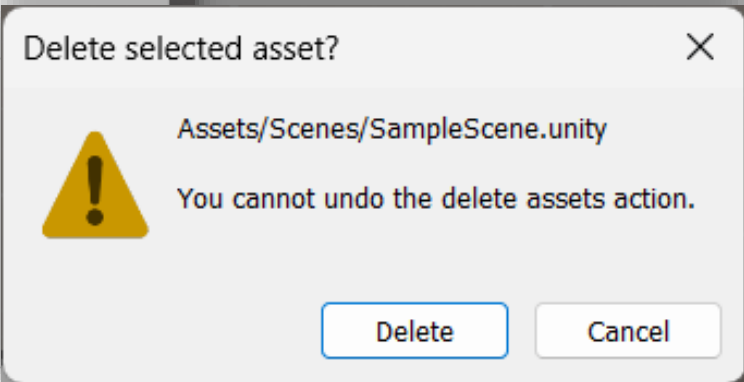
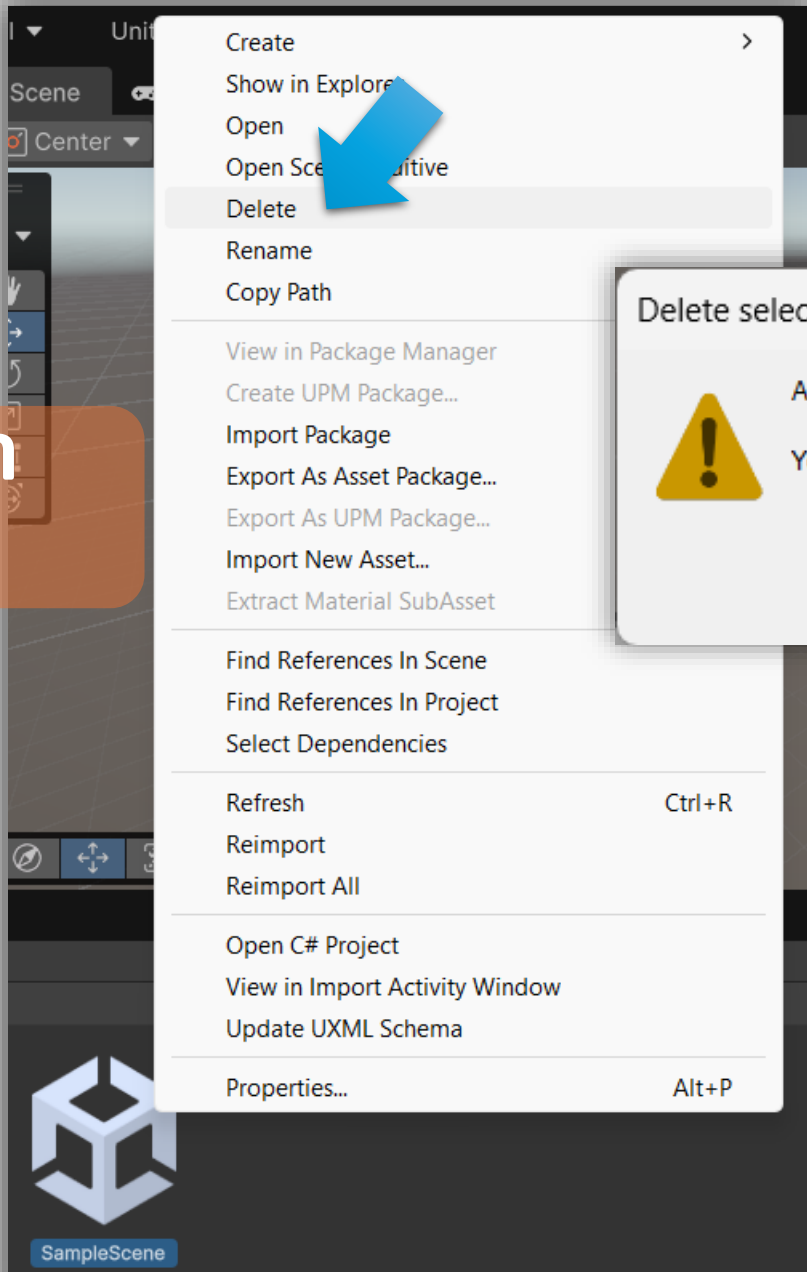


คลิกขวาพื้นที่ว่าง





คลิกขวาจากเท่า
เพื่อลบ



Project Console

+ ▾


★ Favorites

- 🔍 All Materials
- 🔍 All Models
- 🔍 All Prefabs

Assets

- Scenes
- Settings
- TutorialInfo
- Packages

Assets > Scenes



miniGameNo1Scene

Hierarchy

+ ▾ 🔍 All

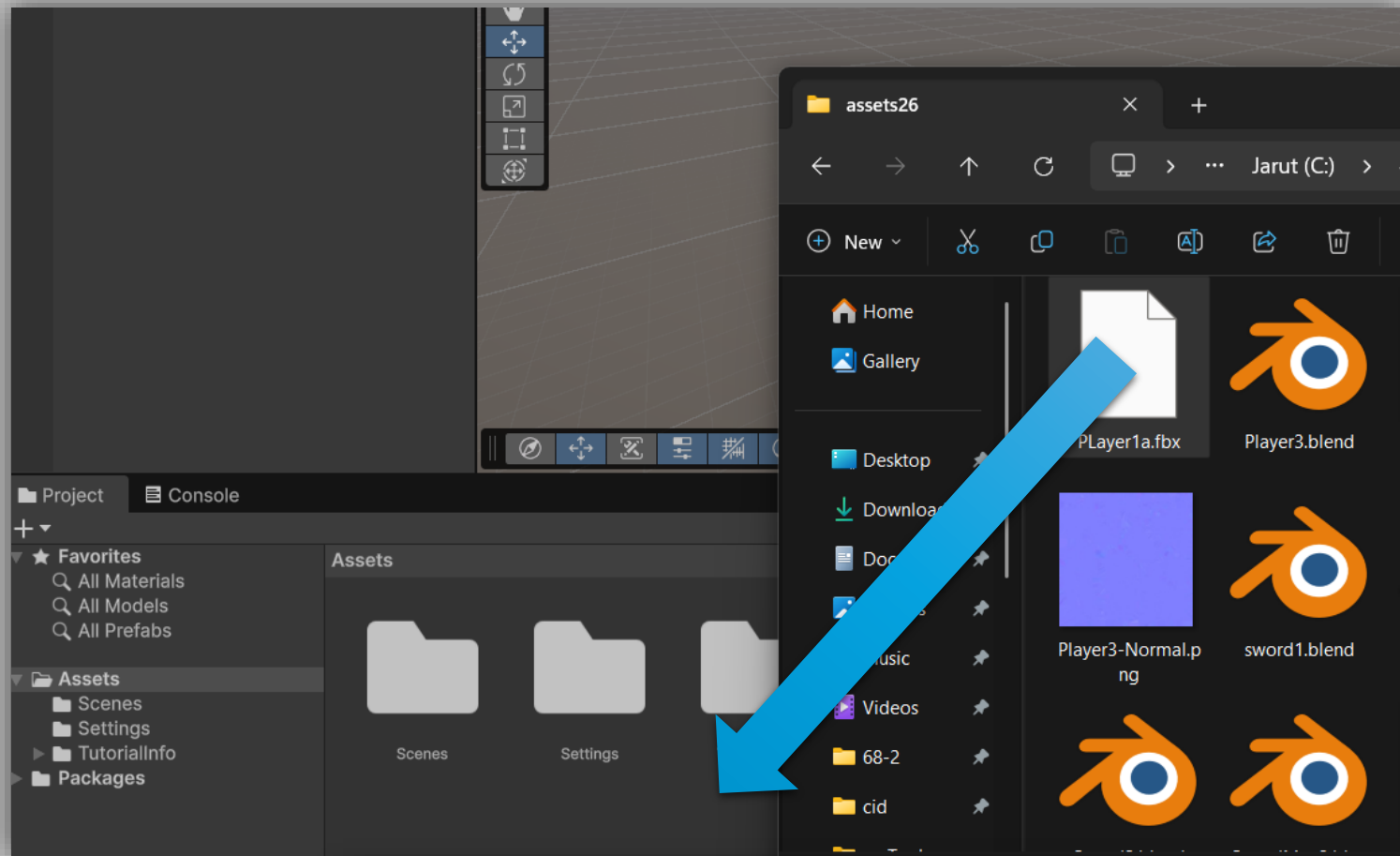
- ▾ 🎮 miniGameNo1Scene
 - 🎮 Main Camera
 - 🎮 Directional Light



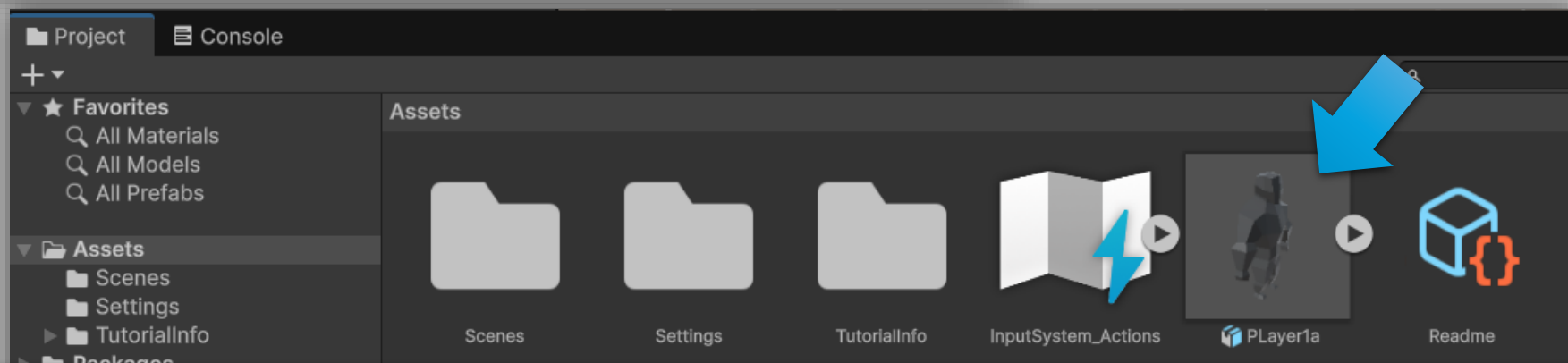
เตรียม Assets

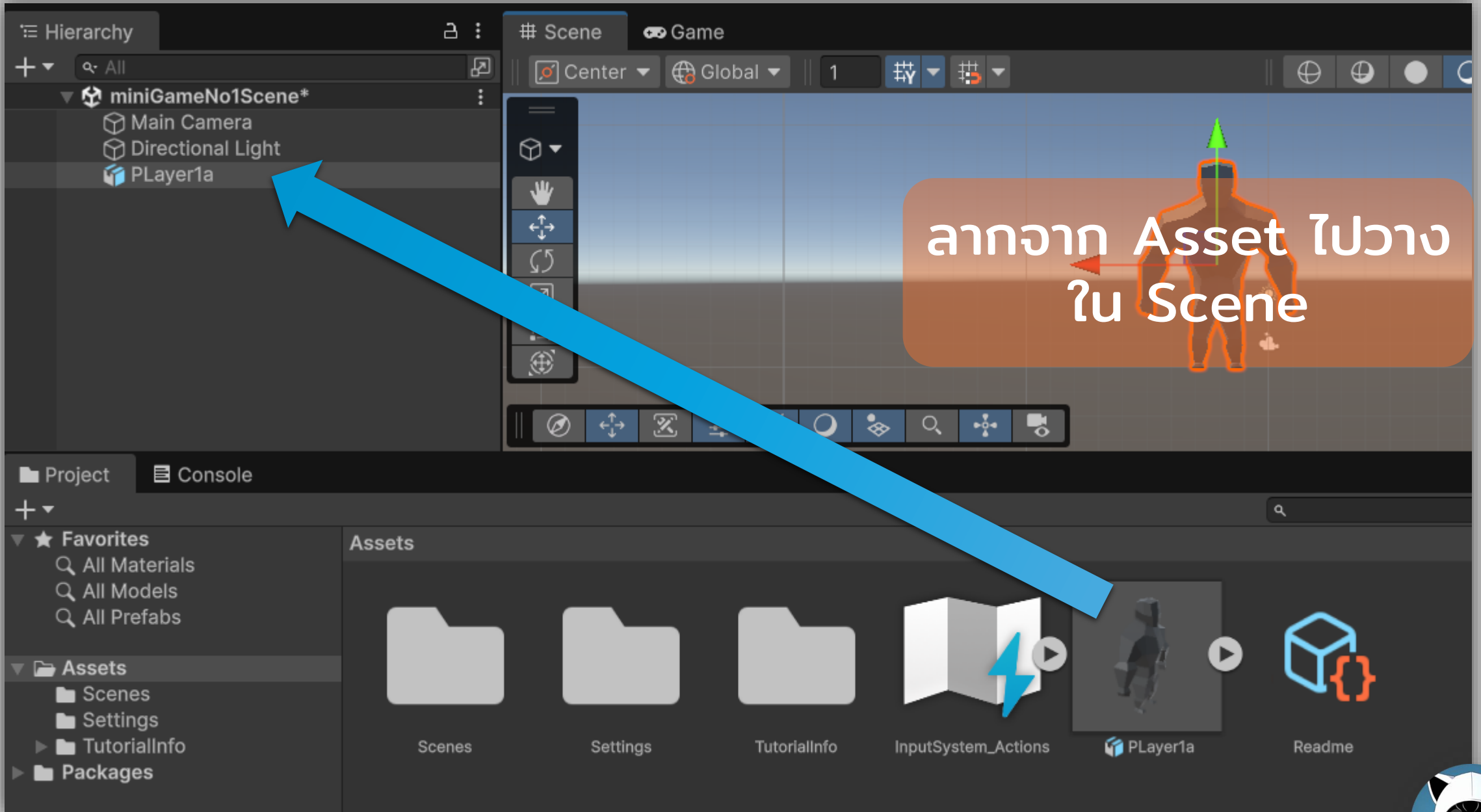
- ผู้เล่น
 - ตัวละครที่ปั้นไว้จาก Blender
 - Tag: Player
- วัตถุ1 (กล่าวถึงทีหลัง)
 - Cube
 - Tag: Enemy1
- วัตถุ2 (กล่าวถึงทีหลัง)
 - Sphere
 - Tag: Enemy2]



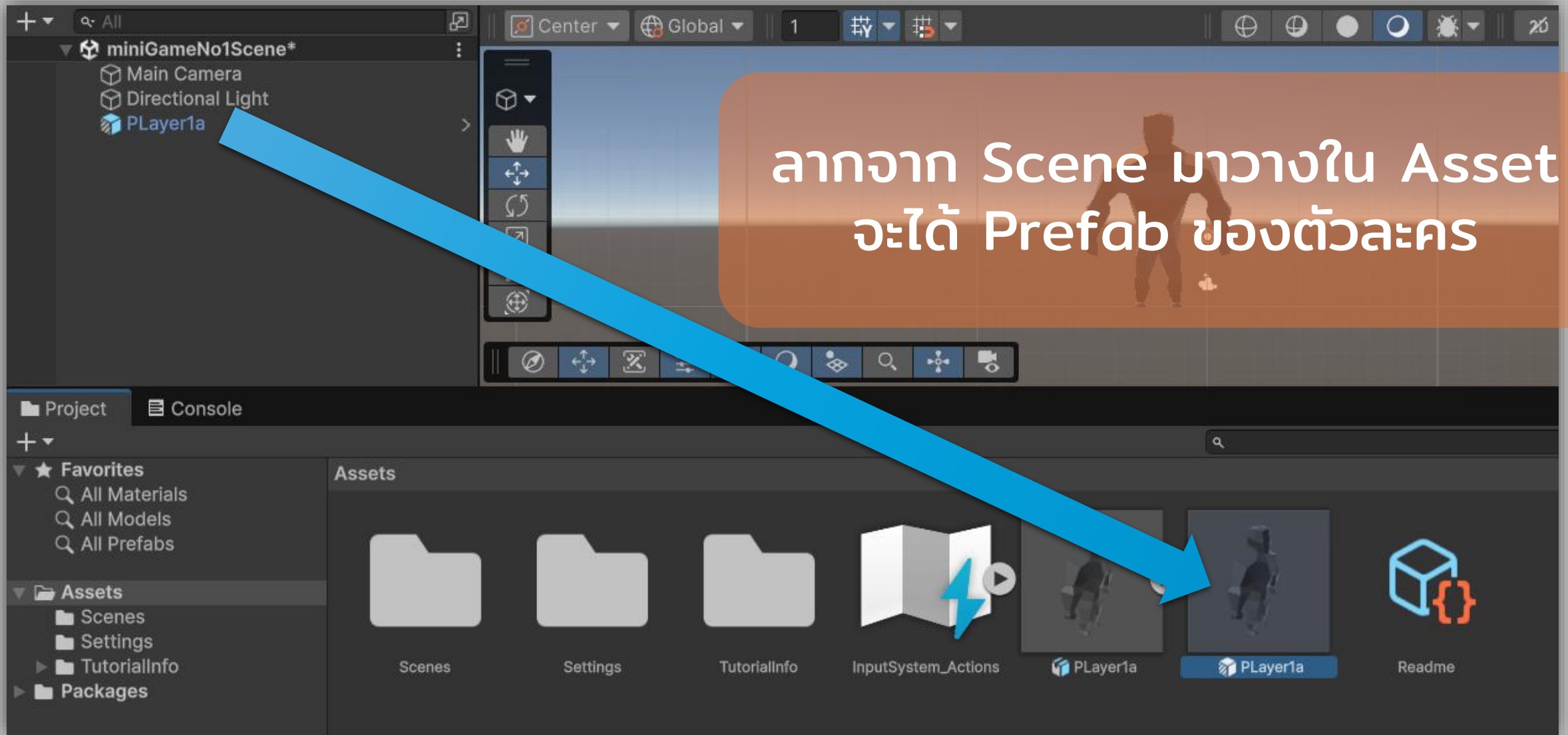


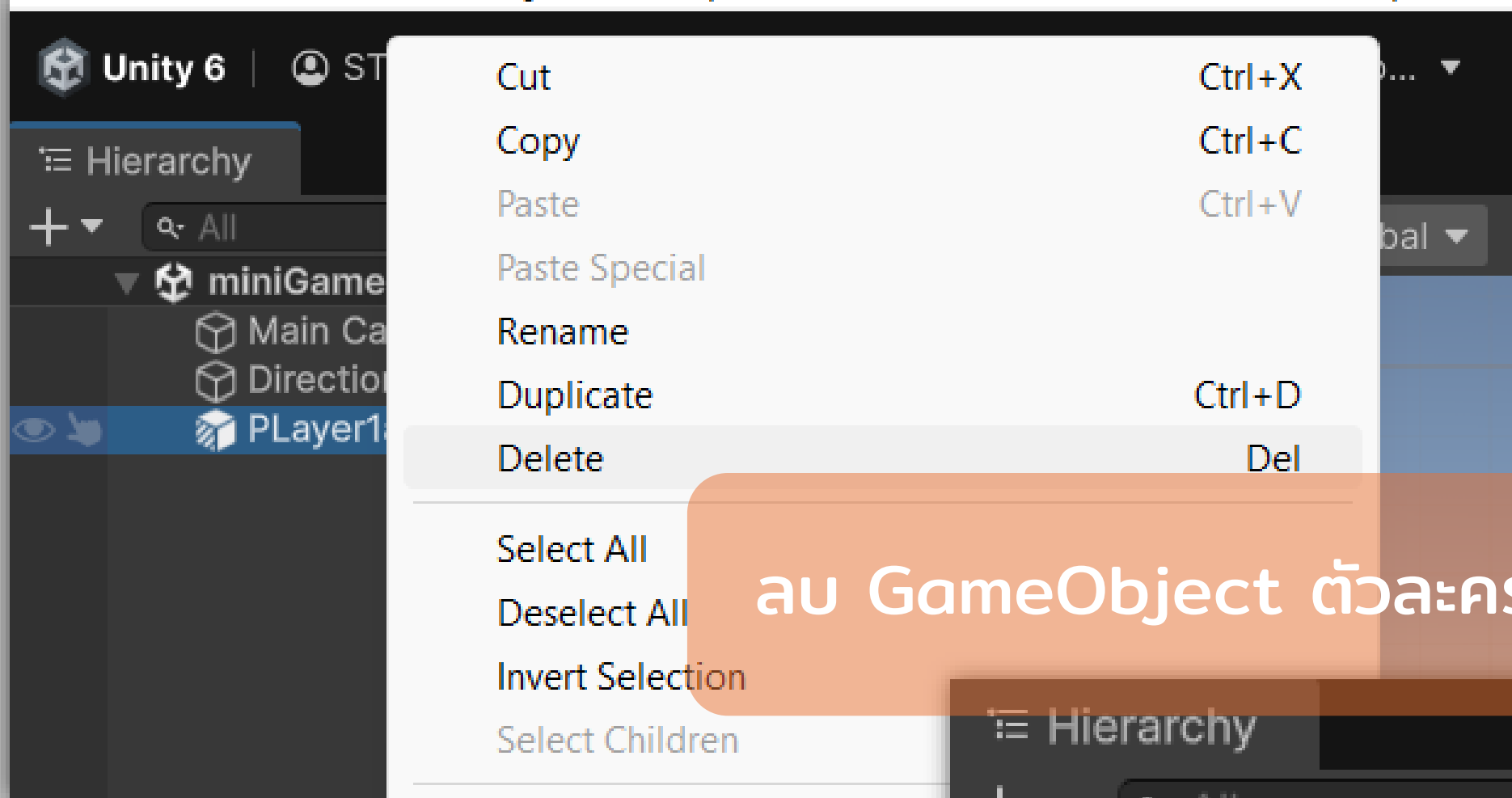
การนำ FBX มาใส่ใน
Assets



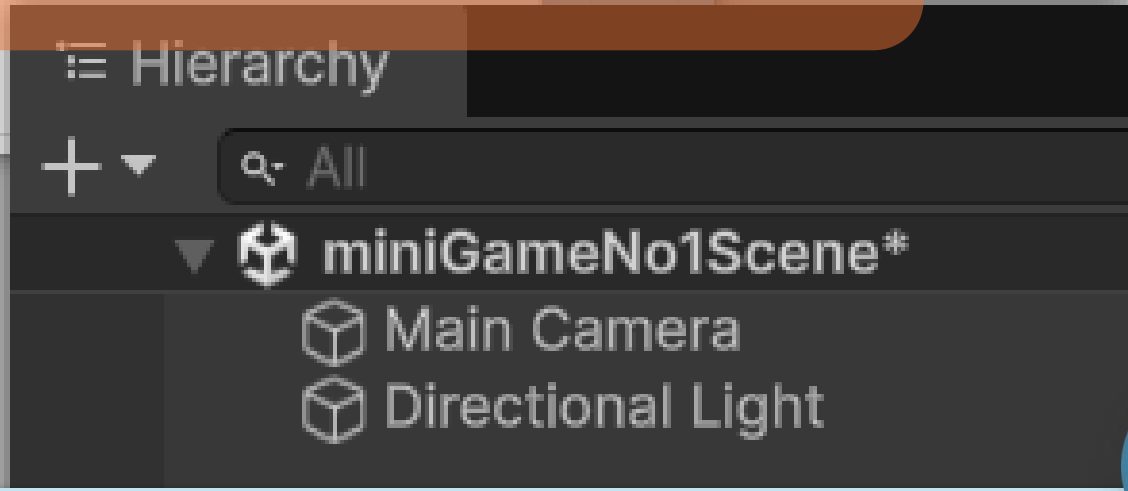


ลากจาก Scene มาวางใน Asset
จะได้ Prefab ของตัวละคร



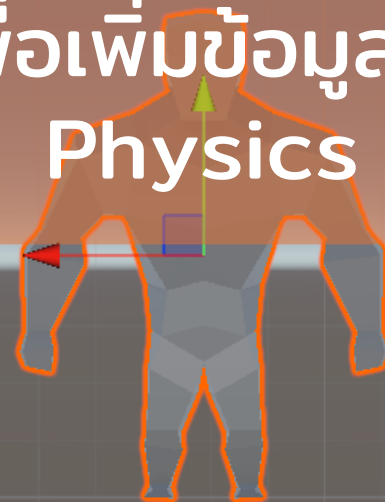


ลบ GameObject ตัวละครทิ้ง



ลูกศรตรงนี้เอาไว้กด
เพื่อกลับไป Scene

Double Click เข้า Prefab ของตัว
ละคร เพื่อเพิ่มข้อมูลเกี่ยวกับ
Physics



Inspector

Player1a Static

Tag Untagged Layer Default

Prefab PLayer1a

Overrides Select Open

Transform

Position X 2.11974 Y 0 Z 2.11974

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Player 1 (Mesh Filter)

Mesh Player1

Mesh Renderer

Materials 1

Lighting

Probes

Additional Settings

Lit (Material)

Shader Universal Re Edit...

Add Component

Lit (Material)

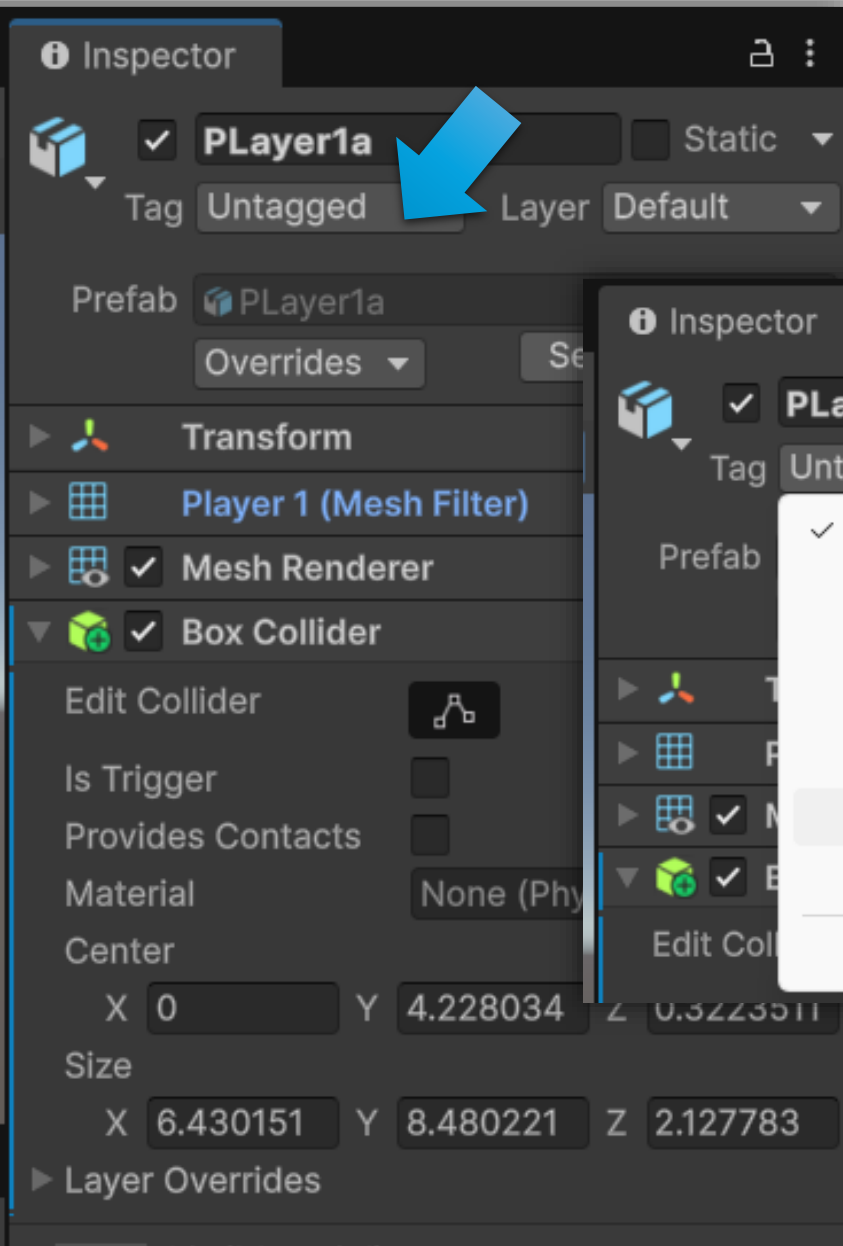
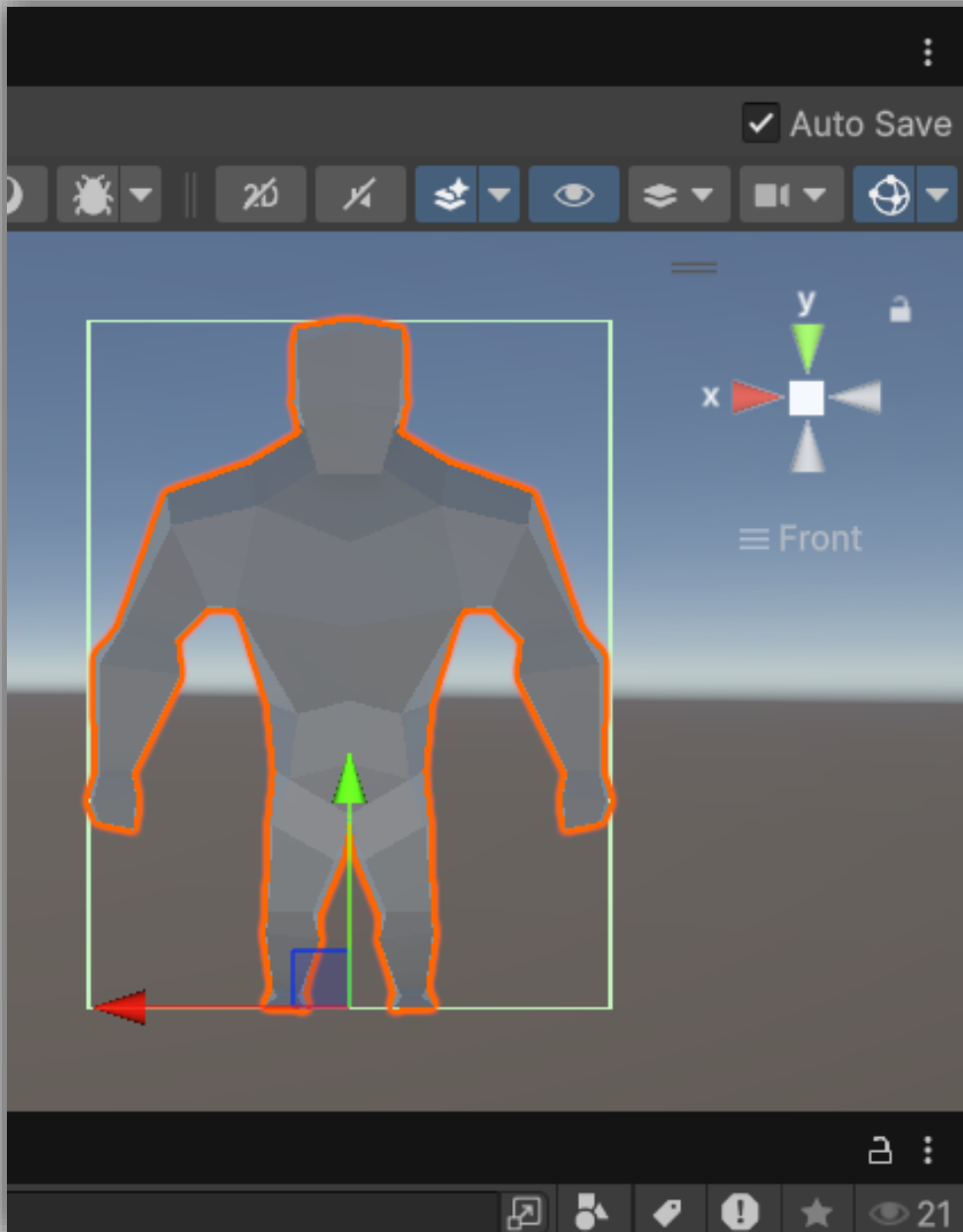
Component

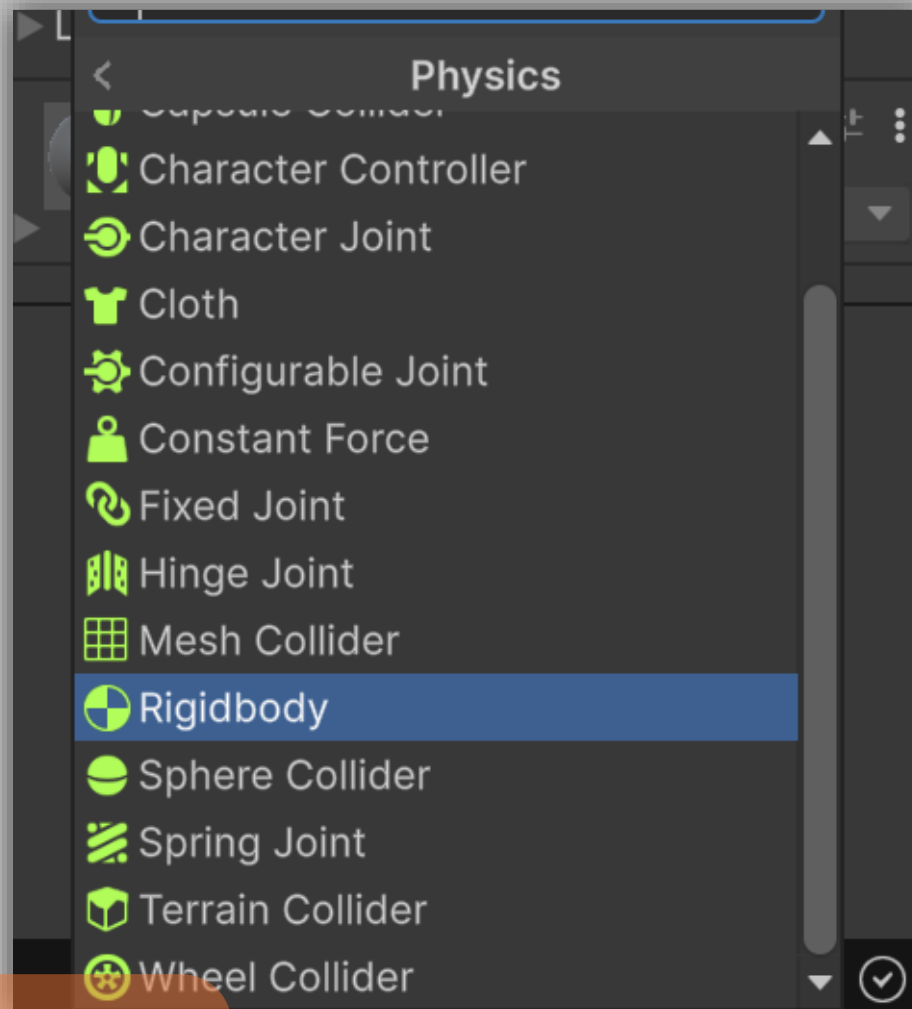
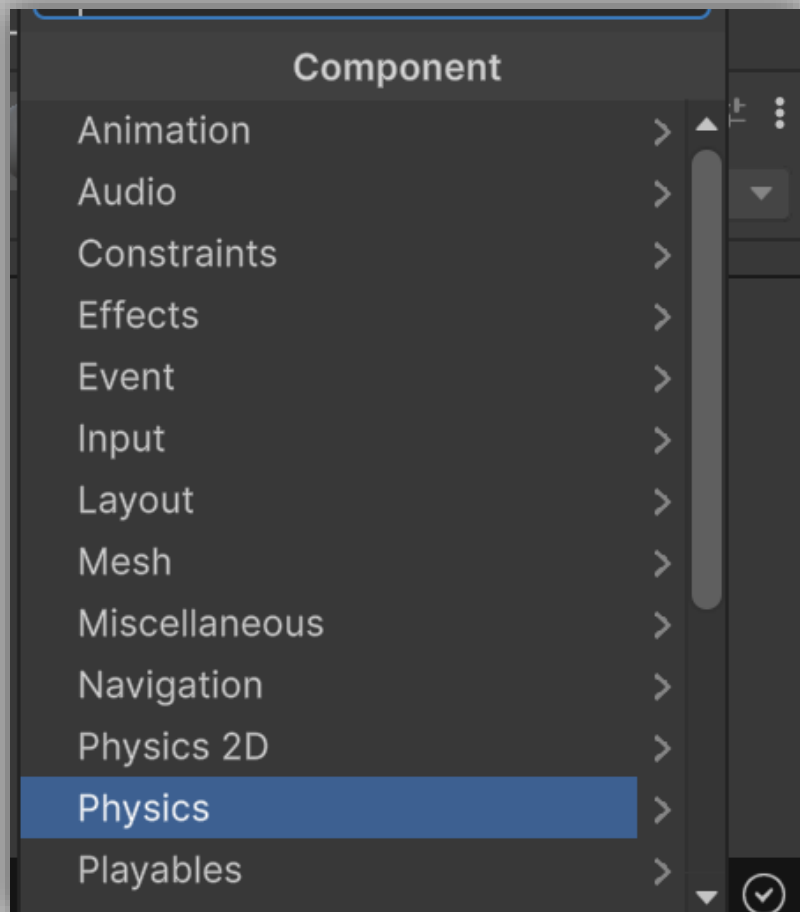
- Effects
- Event
- Input
- Layout
- Mesh
- Miscellaneous
- Navigation
- Physics 2D
- Physics**
- Playables
- Rendering
- Scripts
- Tilemap

Physics

- Articulation Body
- Box Collider**
- Capsule Collider
- Character Controller
- Character Joint
- Cloth
- Configurable Joint
- Constant Force
- Fixed Joint
- Hinge Joint
- Mesh Collider
- Rigidbody
- Sphere Collider







เพิ่ม Component ชื่อ
Rigidbody จาก Physics



จัดจากเกม

วาง Player เพิ่มพื้นที่ และปรับมุมมองกล้อง



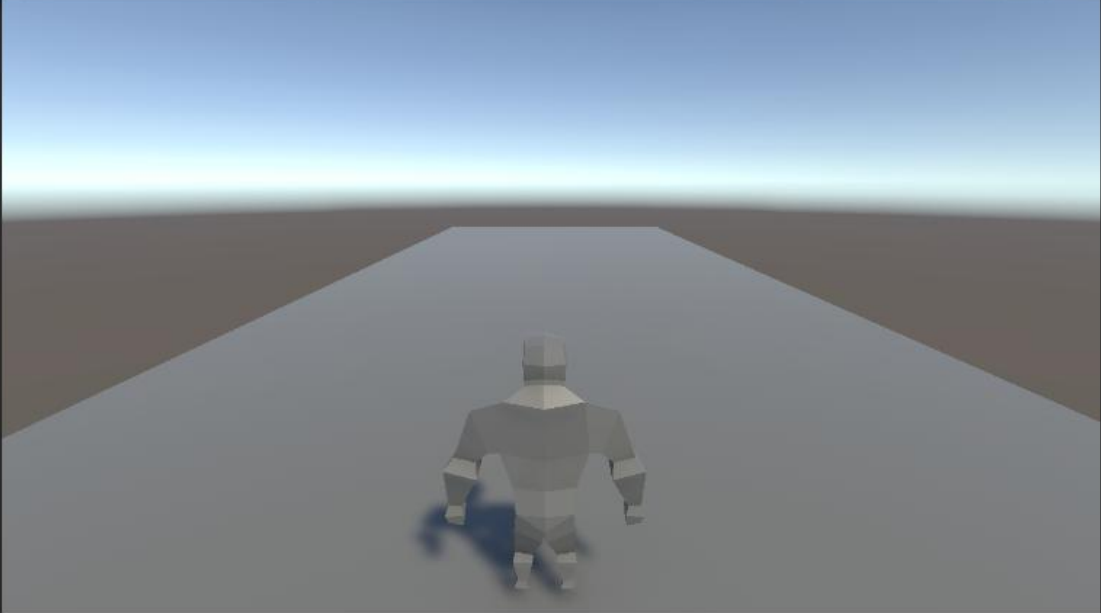
Hierarchy

+ All

- miniGameNo1Scene
 - Main Camera
 - Directional Light
 - Player1a
 - Floor

Game

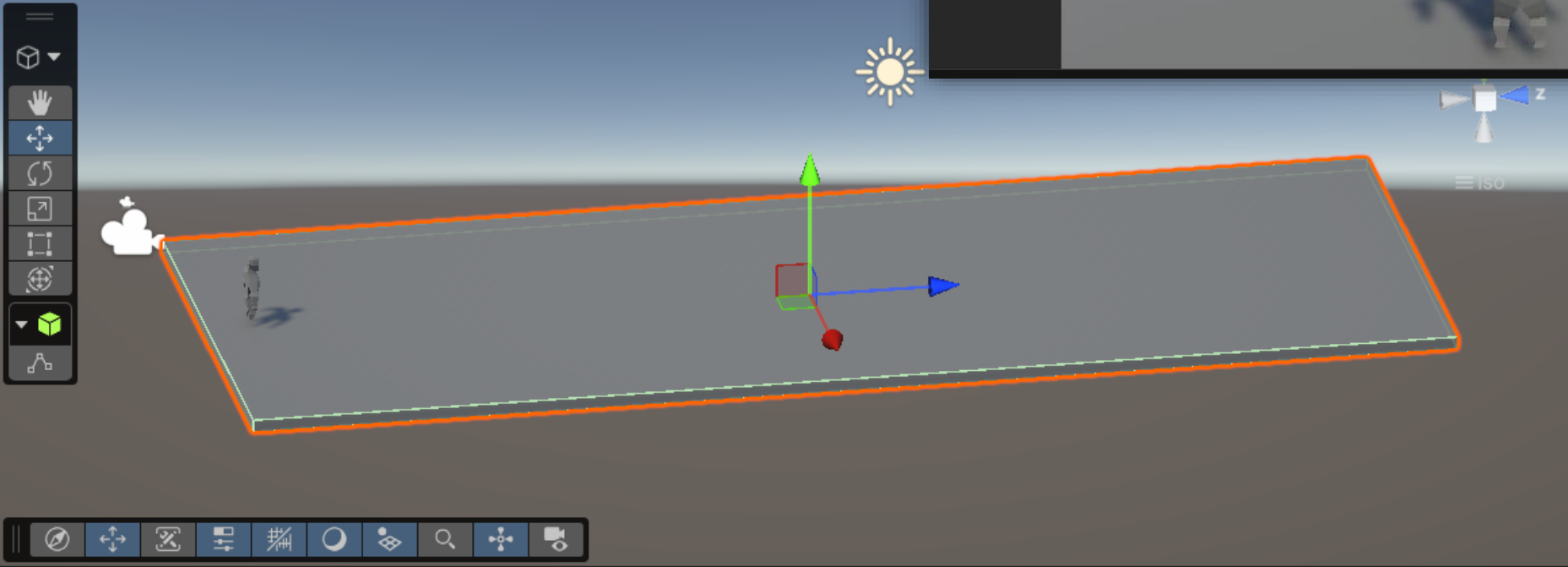
Display 1 Full HD (1920x1080) Scale 0.39x Play Focused



A first-person perspective view of a game. A grey, low-poly character is standing on a grey rectangular platform. The background is a simple landscape with a blue sky and a horizon line.

Scene

Pivot Local 1



A top-down view of the same 3D scene. A grey character is on a grey floor. The floor is outlined with a thick orange border. A 3D coordinate system with red, green, and blue axes is visible in the center. A sun icon is in the top right, and a camera control icon is in the bottom right.



กำหนดคุณสมบัติของผู้เล่น



PlayerProperty.cs

1. สร้างสคริปต์ชื่อ PlayerProperty.cs ด้วยการคลิกขวาที่ส่วนของช่อง Asset เลือก Create > MonoBehaviour Script แล้วตั้งชื่อสคริปต์เป็น PlayerProperty (สิ่งที่ต้องระวัง ถ้าเปลี่ยนชื่อภายหลังข้อมูลใน Source Code จะเป็นคลาสชื่อเดียวกับตอนสร้างไฟล์ ดังนั้น ถ้าเปลี่ยนชื่อไฟล์ใน Asset จะต้องเปลี่ยนชื่อใน Source Code ด้วย)
2. Double Click ที่ PlayerProperty เพื่อเข้าสู่โปรแกรม Visual Studio เพื่อแก้ไขโค้ด



Context menu for creating a new asset in Unity:

- Create >
- Show in Explorer
- Open
- Open Scene Additive
- Delete
- Rename
- Copy Path Alt+Ctrl+C

- View in Package Manager
- Create UPM Package...
- Import Package >
- Export As Asset Package...
- Export As UPM Package...
- Import New Asset...
- Extract Material SubAsset

- Find References In Scene
- Find References In Project
- Select Dependencies

- Refresh Ctrl+R
- Reimport
- Reimport All

- Open C# Project
- View in Import Activity Window
- Update UXML Schema

- Properties... Alt+P

Sub-menu for 'Create':

- Folder
- Material
- MonoBehaviour Script**
- Prefab Variant

- 2D >
- Animation >
- Audio >
- Rendering >
- Scene >
- Scripting >

Sub-menu for 'Properties...':

- Custom Font

- D3D12 Settings >

- Input Actions

Asset browser showing three items:

- Player1a**: A 3D model of a character.
- PlayerProperty**: A blue square icon with a green hash symbol (#). A blue arrow points to this icon.
- Readme**: A blue cube icon with an orange curly brace icon next to it.



```
PlayerProperty.cs  # X
Assembly-CSharp  PlayerProperty  Start()
1  using UnityEngine;
2
3  Unity Script | 0 references
4  public class PlayerProperty : MonoBehaviour
5  {
6      // Start is called once before the first execution of Update after the MonoBehaviour is created
7      Unity Message | 0 references
8      void Start()
9      {
10         }
11
12     // Update is called once per frame
13     Unity Message | 0 references
14     void Update()
15     {
16         }
17 }
```

Solution Explorer

Search Solution Explorer (Ctrl+;) 🔍

- Solution 'miniGameNo1' (2 of 2 proj)
- Assembly-CSharp
 - References
 - Assets
 - TutorialInfo
 - C# PlayerProperty.cs
- Assembly-CSharp-Editor



PlayerProperty.cs

Assembly-CSharp

PlayerProperty

```
1      using UnityEngine;
2
3      Unity Script | 0 references
4      public class PlayerProperty : MonoBehaviour
5      {
6          public int hp;
7          public int hpMax;
8          public int score;
9      }
```



```
using UnityEngine;
```

Unity Script (1 asset reference) | 2 references

```
public class PlayerProperty : MonoBehaviour
```

```
{
```

```
    public int hp = 5;
```

```
    public int hpMax = 5;
```

```
    public int score = 0;
```

Unity Message | 0 references

```
private void Awake()
```

```
{
```

```
    gameObject.tag = "Player";
```

```
}
```

```
}
```

เพิ่ม Awake() สำหรับให้ทำงานเมื่อ
สคริปต์ถูกเรียกให้ทำงาน เพื่อให้
มั่นใจว่าไม่ลืมหาค่า tag ด้วยการ
กำหนด tag เป็น Player

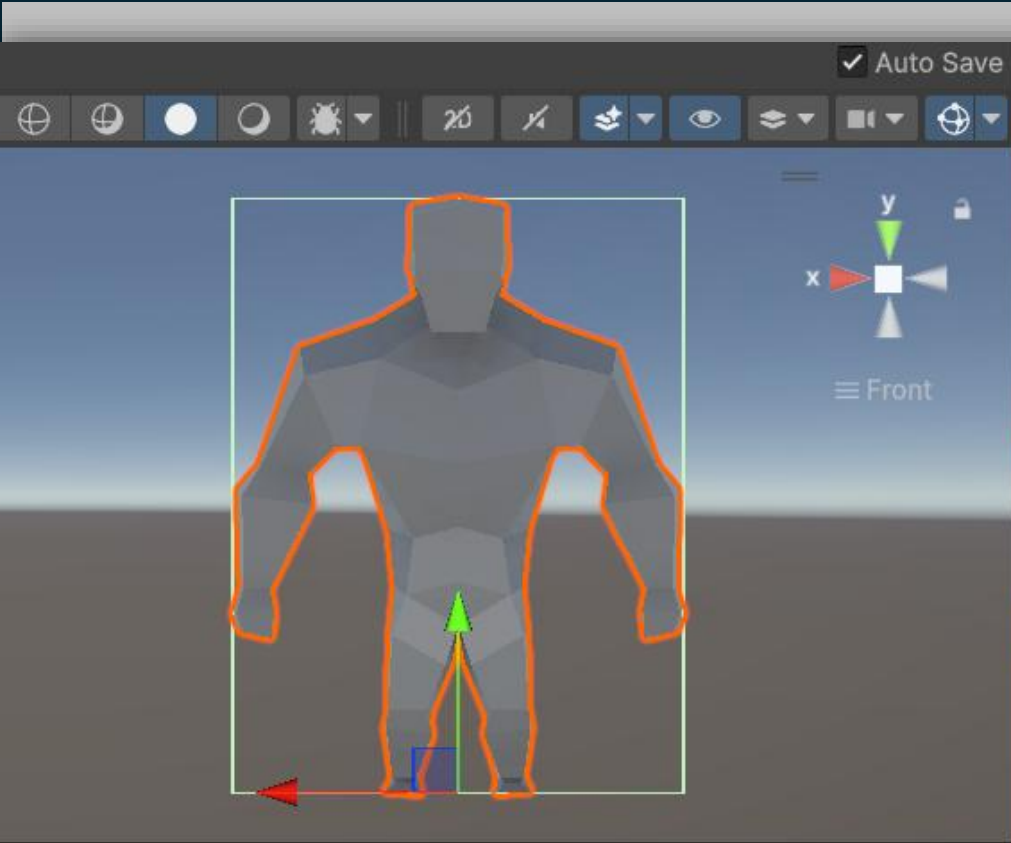


PlayerProperty.cs

3. ເປັນ Prefab ຂອງ Player

4. ລາກ PlayerProperty ໄປວາງໃນຮ່ອງດ້ານຂວາໃນ Inspector ຂອງ Player





Inspector for **Player1a** (Tag: Player, Layer: Default, Static: checked). Prefab: PLayer1a.

Transform

| | | | | | | |
|----------|---|---------|---|-----|---|---------|
| Position | X | 2.11974 | Y | 0 | Z | 2.11974 |
| Rotation | X | 0 | Y | 0 | Z | 0 |
| Scale | X | 0.1 | Y | 0.1 | Z | 0.1 |

Player 1 (Mesh Filter)

Mesh Renderer (checked)

Box Collider (checked)

Edit Collider:

Is Trigger:

Provides Contacts:

Material: None (Physics Materia)

Center

| | | | | | |
|---|---|---|----------|---|-----------|
| X | 0 | Y | 4.228034 | Z | 0.3223511 |
|---|---|---|----------|---|-----------|

Size

| | | | | | |
|---|----------|---|----------|---|----------|
| X | 6.430151 | Y | 8.480221 | Z | 2.127783 |
|---|----------|---|----------|---|----------|

Layer Overrides

Lit (Material) (Shader: Universal Re, Edit...)

Add Component

Inspector for **Player1a** (Tag: Player, Layer: Default, Static: checked). Prefab: PLayer1a.

Transform

Player 1 (Mesh Filter)

Mesh Renderer (checked)

Box Collider (checked)

Player Property (Script) (checked) (Script: PlayerProperty)

Script: **PlayerProperty**

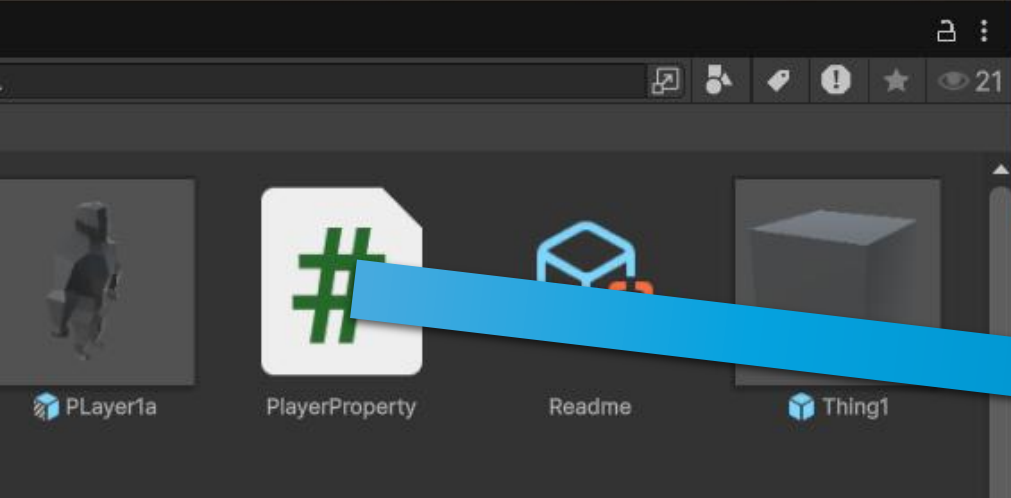
Hp: 0

Hp Max: 0

Score: 0

Lit (Material) (Shader: Universal Re, Edit...)

Add Component



PlayerProperty.cs

5. ลองปรับโค้ดในคลาสPlayerProperty ให้

hp = 5

hpMax = 5

score = 0

6. ทำการอัปเดตด้วยการกด Reset จาก Inspector



```
PlayerProperty.cs [ + ] [ x ]
Assembly-CSharp [ v ] PlayerProperty
1 using UnityEngine;
2
3 Unity Script (1 asset reference) | 0 references
4 public class PlayerProperty : MonoBehaviour
5 {
6     public int hp = 5;
7     public int hpMax = 5;
8     public int score = 0;
9 }
```

Player Property (Script)

- Script
- Hp
- Hp Max
- Score

- Reset
- Added Component >
- Remove Component
- Move Up
- Move Down
- Copy Component
- Paste Component As New
- Paste Component Values
- Find References In Scene
- Properties...
- Edit Script

Player Property (Script)

| | |
|--------|----------------|
| Script | PlayerProperty |
| Hp | 5 |
| Hp Max | 5 |
| Score | 0 |



การควบคุมผู้เล่น



PlayerActivity.cs

1. สร้างสคริปต์ชื่อ PlayerActivity.cs ด้วยการคลิกขวาที่ส่วนของช่อง Asset เลือก Create > MonoBehaviour Script แล้วตั้งชื่อสคริปต์เป็น PlayerActivity
2. Double Click ที่ PlayerActivity เพื่อเข้าสู่โปรแกรม Visual Studio เพื่อแก้ไขโค้ด
3. เข้า Prefab ของ Player
4. ลาก PlayerActivity ไปวางในช่องด้านขวาใน Inspector ของ Player



Context menu for a 3D model in Unity:

- Create >
- Show in Explorer
- Open
- Open Scene Additive
- Delete
- Rename
- Copy Path Alt+Ctrl+C
- View in Package Manager
- Create UPM Package...
- Import Package >
- Export As Asset Package...
- Export As UPM Package...
- Import New Asset...
- Extract Material SubAsset
- Find References In Scene
- Find References In Project
- Select Dependencies
- Refresh Ctrl
- Reimport
- Reimport All
- Open C# Project
- View in Import Activity Window
- Update UXML Schema
- Properties... Alt+P

Sub-menu for 'Create':

- Folder
- Material
- MonoBehaviour Script**
- Prefab Variant
- 2D >
- Animation >
- Audio >
- Rendering >
- Scene >
- Scripting >
- Shader >

Bottom sub-menu:

- D3D12 Settings >
- Input Actions



Asset browser showing three items:

- Player1a**: A 3D model of a character.
- PlayerActivity**: A blue square icon with a green hash symbol (#). A blue arrow points to this icon.
- PlayerProperty**: A white square icon with a green hash symbol (#).



PlayerActivity.cs

PlayerProperty.cs

Assembly-CSharp

PlayerActivity

Start()

```
1 using UnityEngine;
2
3 public class PlayerActivity : MonoBehaviour
4 {
5     // Start is called once before the first execution of Update after the MonoBehaviour is created
6     void Start()
7     {
8     }
9
10
11     // Update is called once per frame
12     void Update()
13     {
14     }
15 }
16
17
```



```
PlayerActivity.cs x PlayerProperty.cs
Assembly-CSharp PlayerActivity
1 using UnityEngine;
2 public class PlayerActivity : MonoBehaviour
3 {
4     void Start()
5     {
6     }
7
8     void Update()
9     {
10    }
11
12    void MoveLeft()
13    {
14    }
15
16    void MoveRight()
17    {
18    }
19
20    void IncScore()
21    {
22    }
23
24    void DecScore()
25 }
```

สร้าง method

1. MoveLeft()
2. MoveRight()
3. IncScore()
4. DecScore()

ของเดิมที่มีอยู่

1. Start() ทำงานครั้งเดียวเมื่อสคริปต์ถูกเรียก
2. Update() ทำงานทุกเฟรมที่มีการเรนเดอร์ภาพ



Inspector

Player1a Static

Tag **Player** Layer **Default**

Prefab **Player1a**

Overrides

Transform

Player 1 (Mesh Filter)

Mesh Renderer

Box Collider

Player Property (Script)

Script **PlayerProperty**

Hp 5

Hp Max 5

Score 0

Player Activity (Script)

Script **PlayerActivity**

Lit (Material)

Shader **Universal Re**



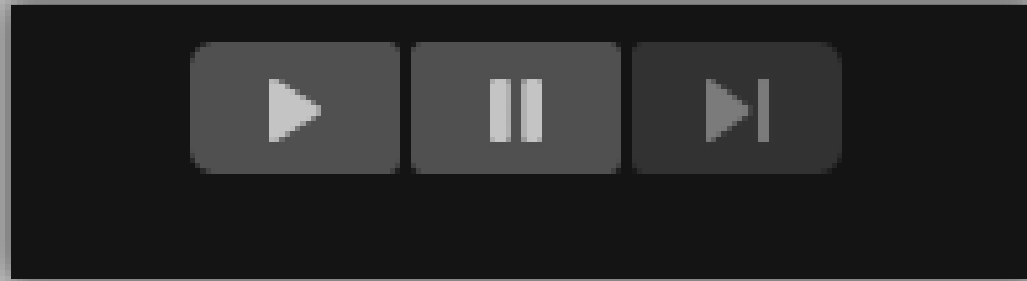
```
public float speed = 5.0f;
```

```
Unity Message | 0 references
```

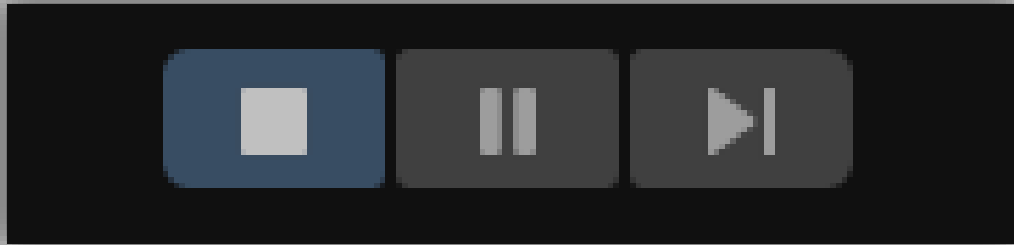
```
void Update()
```

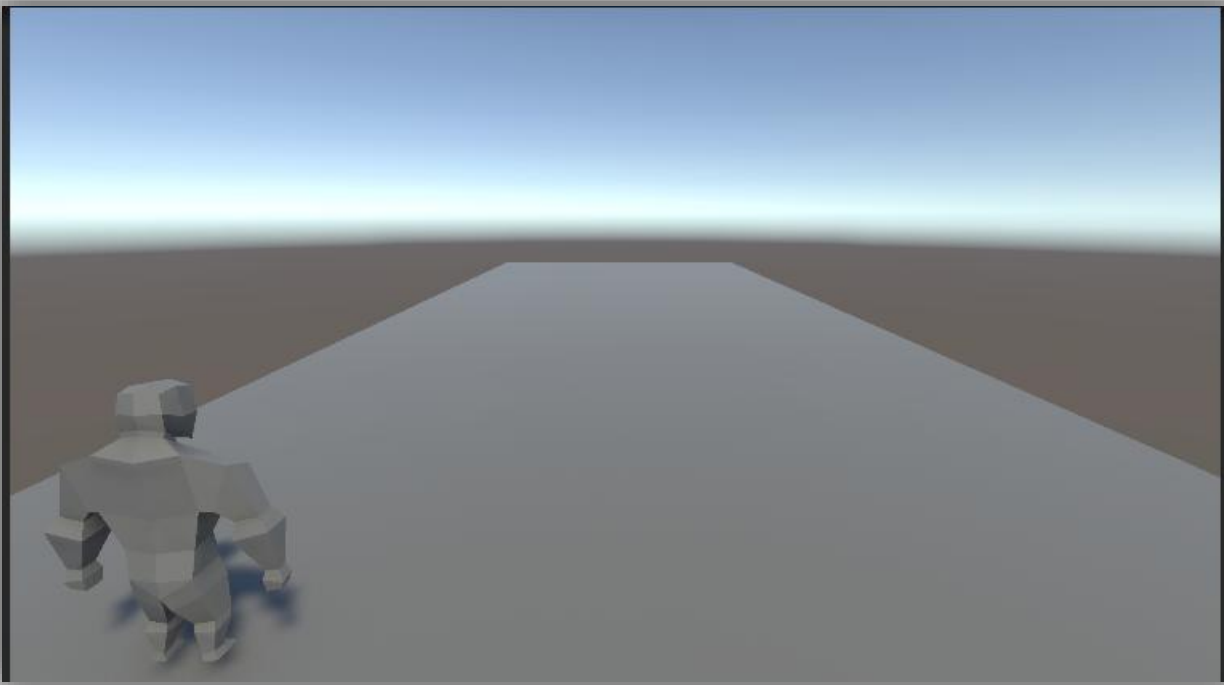
```
{  
    if (Keyboard.current.aKey.isPressed)  
    {  
        Vector3 direction = Vector3.left * speed * Time.deltaTime;  
        transform.Translate(direction);  
    }  
    if (Keyboard.current.dKey.isPressed)  
    {  
        Vector3 direction = Vector3.right * speed * Time.deltaTime;  
        transform.Translate(direction);  
    }  
    Debug.Log(transform.position);  
}
```



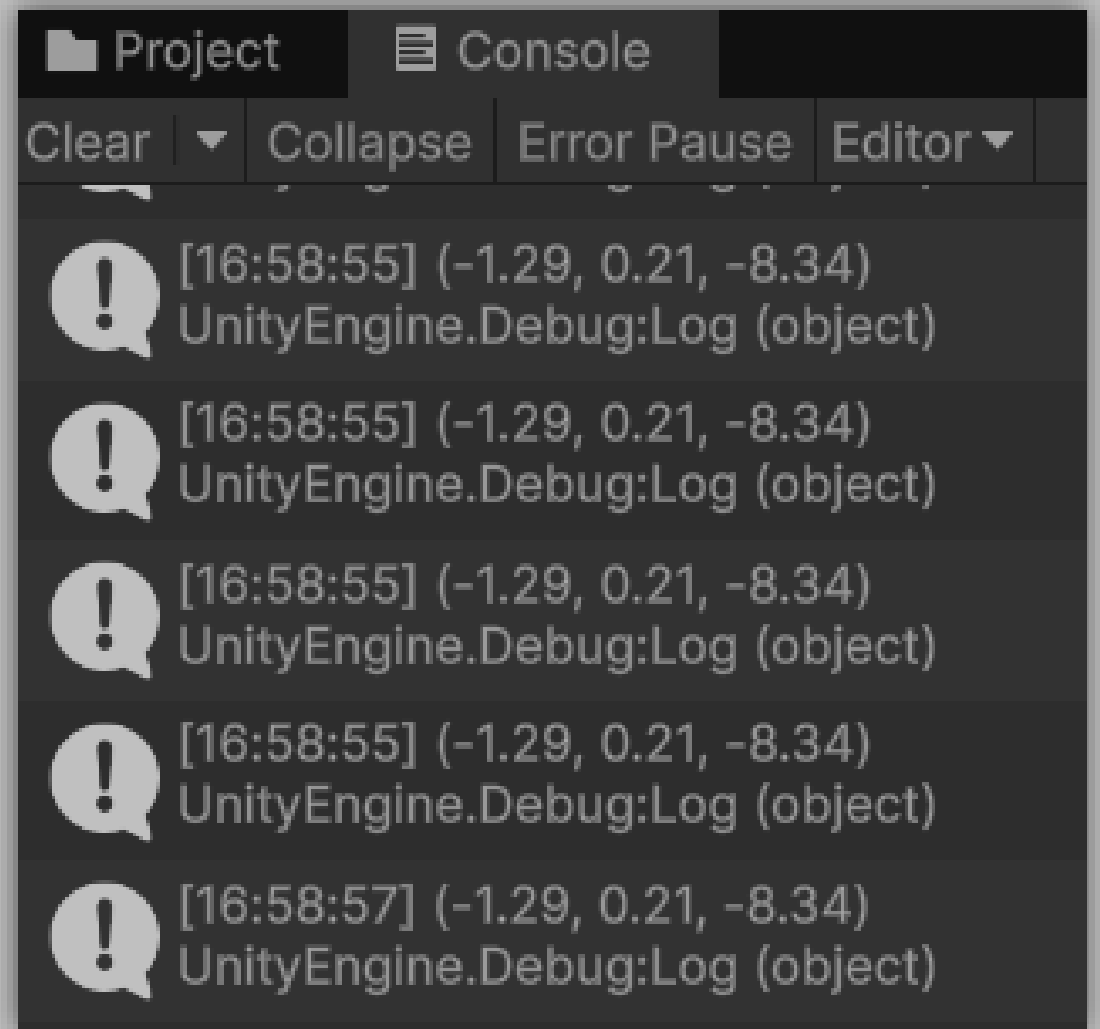


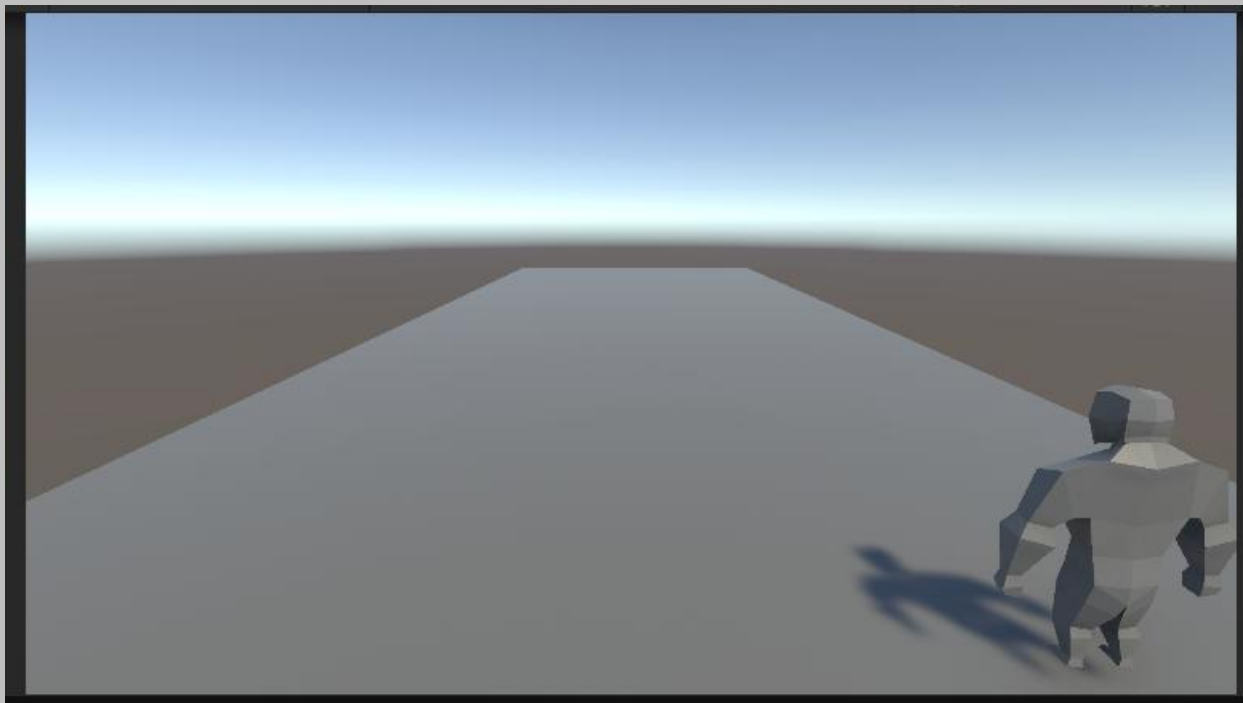
คำสั่งเล่น/หยุด พัก



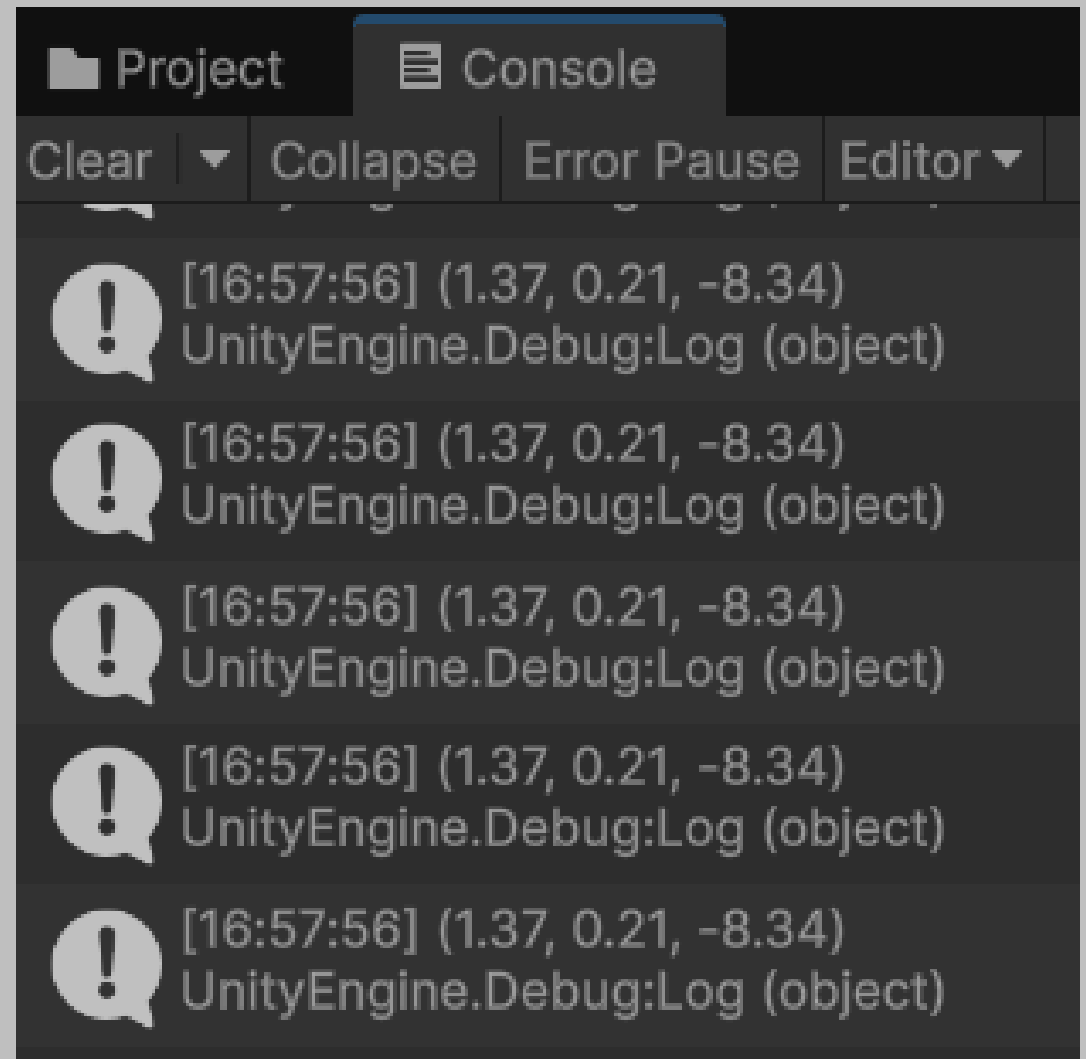


กด Play แล้วขยับตัวละครไปทางซ้ายเพื่อ
ดูว่าค่า x เป็นเท่าไร
จากตัวอย่างพบว่า $x = -1.29$ เป็นตำแหน่ง
ที่เหมาะสมสำหรับขอบจอด้านซ้าย





กด Play แล้วขยับตัวละครไปทางซ้ายเพื่อ
ดูว่าค่า x เป็นเท่าไร
จากตัวอย่างพบว่า $x=1.29$ เป็นตำแหน่งที่
เหมาะสมสำหรับขอบจอด้านขวา
(จะได้เท่า ๆ กับด้านซ้ายที่เป็น -1.29)



เดินซ้าย/ขวาไม่ออกนอกจอแล้วละ

```
void Update()
{
    Vector3 curPos = transform.position;
    if (Keyboard.current.aKey.isPressed)
    {
        Vector3 direction = Vector3.left * speed * Time.deltaTime;
        curPos += direction;
        if (curPos.x > -1.29)
        {
            transform.Translate(direction);
        }
    }
    if (Keyboard.current.dKey.isPressed)
    {
        Vector3 direction = Vector3.right * speed * Time.deltaTime;
        curPos += direction;
        if (curPos.x < 1.29)
        {
            transform.Translate(direction);
        }
    }
}
```



ปรับปรุงโค้ดใหม่

```
public float speed = 5.0f;
```

```
☒ Unity Message | 0 references
```

```
void Update()
```

```
{  
    if (Keyboard.current.aKey.isPressed)  
    {  
        MoveLeft();  
    }  
    if (Keyboard.current.dKey.isPressed)  
    {  
        MoveRight();  
    }  
}
```

```
void MoveLeft()
```

```
{  
    Vector3 curPos = transform.position;  
    Vector3 direction = Vector3.left * speed * Time.deltaTime;  
    curPos += direction;  
    if (curPos.x > -1.29)  
    {  
        transform.Translate(direction);  
    }  
}
```

```
void MoveRight()
```

```
{  
    Vector3 curPos = transform.position;  
    Vector3 direction = Vector3.right * speed * Time.deltaTime;  
    curPos += direction;  
    if (curPos.x < 1.29)  
    {  
        transform.Translate(direction);  
    }  
}
```



การเข้าถึง

- ผู้เล่น
- คุณสมบัติ
- พฤติกรรม



การอ้างอิงถึงวัตถุ (ผู้เล่น)

- เข้าถึงผู้เล่นต้องสร้างตัวแปรประเภทเดียวกับผู้เล่น

```
GameObject player;
```

- ทำการค้นหาวัตถุที่ตั้งแท็กเป็น Player

```
player = GameObject.FindWithTag("Player");
```

ถ้า player เป็น null แสดงว่าหาวัตถุที่tag
ว่า Player ไม่พบ



การอ้างอิงถึง PlayerProperty

1. เข้าถึงผู้เล่นต้องสร้างตัวแปรประเภท PlayerProperty (ชื่อเดียวกับคลาสที่สร้างไว้)

```
PlayerProperty stats;
```

2. ทำการเข้าถึง PlayerProperty

```
if (player != null)
```

```
{
```

```
    stats = player.GetComponent<PlayerProperty>();
```

```
}
```



การอ้างอิงถึง PlayerActivity

1. เข้าถึงผู้เล่นต้องสร้างตัวแปรประเภท PlayerActivity (ชื่อเดียวกับคลาสที่สร้างไว้)

```
PlayerActivity acts;
```

2. ทำการเข้าถึง PlayerActivity

```
if (player != null)
```

```
{
```

```
    act = player.GetComponent<PlayerActivity>();
```

```
}
```



```
private GameObject player;  
private PlayerProperty stats;
```

Unity Message | 0 references

```
void Start()
```

```
{
```

```
    player = GameObject.FindWithTag("Player");
```

```
    if (player != null)
```

```
    {
```

```
        stats = player.GetComponent<PlayerProperty>();
```

```
    }
```

```
}
```



```
void DecScore()
{
    stats.score -= 1;
    if (stats.score < 0)
    {
        stats.hp -= 1;
        stats.score = 0;
    }
}
```



```
void IncScore()  
{  
    stats.score += 1;  
}
```



อธิบาย Vector





Vector3

 Unity 6.0



Vector3 ใน Unity เป็นโครงสร้างข้อมูล (struct)
ที่ใช้แทนตำแหน่งในปริภูมิสามมิติ (3D space)



ซึ่งประกอบด้วยค่าสามค่าหลัก ได้แก่

x: ค่าที่แสดงถึงตำแหน่งตามแกน X (แกนแนวนอน)

y: ค่าที่แสดงถึงตำแหน่งตามแกน Y (แกนแนวตั้ง)

z: ค่าที่แสดงถึงตำแหน่งตามแกน Z (แกนลึก หรือข้างหน้า/ข้างหลัง)



- กำหนดตำแหน่งของ GameObject ในฉาก เช่น ตำแหน่งเริ่มต้น ตำแหน่งปลายทางของการเคลื่อนที่
- ใช้ในการคำนวณระยะทาง ทิศทาง ความเร็ว และปริมาณทางฟิสิกส์อื่น ๆ
- ใช้ในการแปลงตำแหน่งจากระบบพิกัดหนึ่งไปยังอีกระบบพิกัดหนึ่ง



```
// สร้าง Vector3 เพื่อกำหนดตำแหน่ง
Vector3 myPosition = new Vector3(10, 5, 2);

// กำหนดตำแหน่งให้กับ GameObject
GameObject myGameObject = GameObject.Find("MyObject");
myGameObject.transform.position = myPosition;

// คำนวณระยะห่างระหว่างสองตำแหน่ง
Vector3 positionA = new Vector3(0, 0, 0);
Vector3 positionB = new Vector3(3, 4, 5);
float distance = Vector3.Distance(positionA, positionB);
```



```
// การเคลื่อนที่  
// เคลื่อนที่ไปข้างหน้า  
transform.Translate(Vector3.forward * Time.deltaTime);
```



```
// การหมุน  
// หมุนรอบแกน Y  
transform.Rotate(Vector3.up * Time.deltaTime * 100);
```



```
// การปรับขนาด  
// เพิ่มขนาดเป็นสองเท่า  
transform.localScale = new Vector3(2, 2, 2);
```



Vector3.zero

แทน Vector3 ที่มีค่า x, y, z เป็น 0 ทั้งหมด มักใช้ในการตั้งค่าเริ่มต้นของตำแหน่ง หรือในการเปรียบเทียบว่า Vector3 ใดมีค่าเป็น 0

Vector3.one

แทน Vector3 ที่มีค่า x, y, z เป็น 1 ทั้งหมด มักใช้ในการปรับขนาดของวัตถุให้เท่ากันในทุกแกน

Vector3.forward

แทน Vector3 ที่ชี้ไปข้างหน้าตามแกน Z (0, 0, 1) มักใช้ในการเคลื่อนที่ไปข้างหน้าของวัตถุ



Vector3.back

แกน Vector3 ที่ชี้ไปข้างหลังตามแกน Z (0, 0, -1) มักใช้ในการเคลื่อนที่ไปข้างหลังของวัตถุ

Vector3.up

แกน Vector3 ที่ชี้ขึ้นบนตามแกน Y (0, 1, 0) มักใช้ในการเคลื่อนที่ขึ้นบนของวัตถุ

Vector3.down

แกน Vector3 ที่ชี้ลงล่างตามแกน Y (0, -1, 0) มักใช้ในการเคลื่อนที่ลงล่างของวัตถุ



Vector3.left

แกน Vector3 ที่ชี้ไปทางซ้ายตามแกน X $(-1, 0, 0)$ มักใช้ในการเคลื่อนที่ไปทางซ้ายของวัตถุ

Vector3.right

แกน Vector3 ที่ชี้ไปทางขวาตามแกน X $(1, 0, 0)$ มักใช้ในการเคลื่อนที่ไปทางขวาของวัตถุ



```
// เคลื่อนที่วัตถุไปข้างหน้า 1 หน่วยต่อเฟรม  
transform.Translate(Vector3.forward * Time.deltaTime);  
  
// หมุนวัตถุรอบแกน Y 90 องศา  
transform.Rotate(Vector3.up, 90);  
  
// ปรับขนาดของวัตถุให้ใหญ่ขึ้นเป็นสองเท่าในทุกแกน  
transform.localScale = Vector3.one * 2;
```



$speed * Time.deltaTime$

มีความหมายว่าให้เคลื่อนที่ด้วยความเร็ว
"speed ต่อ วินาที"



UI แสดงสถานะของผู้เล่น



การสร้าง UI ใน Unity 6.3 ด้วยระบบ **uGUI**
หรือ UI (Canvas) เพื่อแสดงค่าจาก Script เป็น
พื้นฐานสำคัญในการทำเกม



ขั้นตอนแบ่งออกเป็น 2 ส่วนหลัก คือ

1. การเตรียม UI

2. การเขียน Code เพื่อเชื่อมโยงข้อมูล



การเตรียม UI ใน Unity Editor

1. สร้าง Canvas ด้วยการคลิกขวาใน Hierarchy > UI > Canvas
2. สร้าง UI Elements
ภายใต้ Canvas ให้สร้าง Object ดังนี้
 1. Slider (สำหรับ HP Bar)
คลิกขวาที่ Canvas > UI > Slider (ตั้งชื่อว่า HPSlider)
 2. Text (สำหรับ Score)
คลิกขวาที่ Canvas > UI > Text - TextMeshPro (ตั้งชื่อว่า ScoreText)
3. ตั้งค่า Slider
ใน Inspector ของ HPSlider ให้ติ๊ก Interactable ออก (เพื่อไม่ให้ผู้เล่นลากแถบเลือดเองได้)



Context menu for creating a new object in Unity:

- Create Empty (Ctrl+Shift+N)
- Create Empty Child (Alt+Shift+N)
- Create Empty Parent (Ctrl+Shift+G)
- 2D Object >
- 3D Object >
- Effects >
- Light >
- Audio >
- Video >
- UI >** (highlighted with a blue arrow)
- AI >
- UI Toolkit >
- Rendering >
- Volume >
- Camera
- Visual Scripting Scene Variables

- Center On Children

- Make Parent
- Clear Parent

- Set as first sibling (Ctrl+=)
- Set as last sibling (Ctrl+-)
- Move To View (Ctrl+Alt+F)
- Align With View (Ctrl+Shift+F)

Sub-menu for the UI category:

- Image
- Text - TextMeshPro
- Raw Image
- Panel

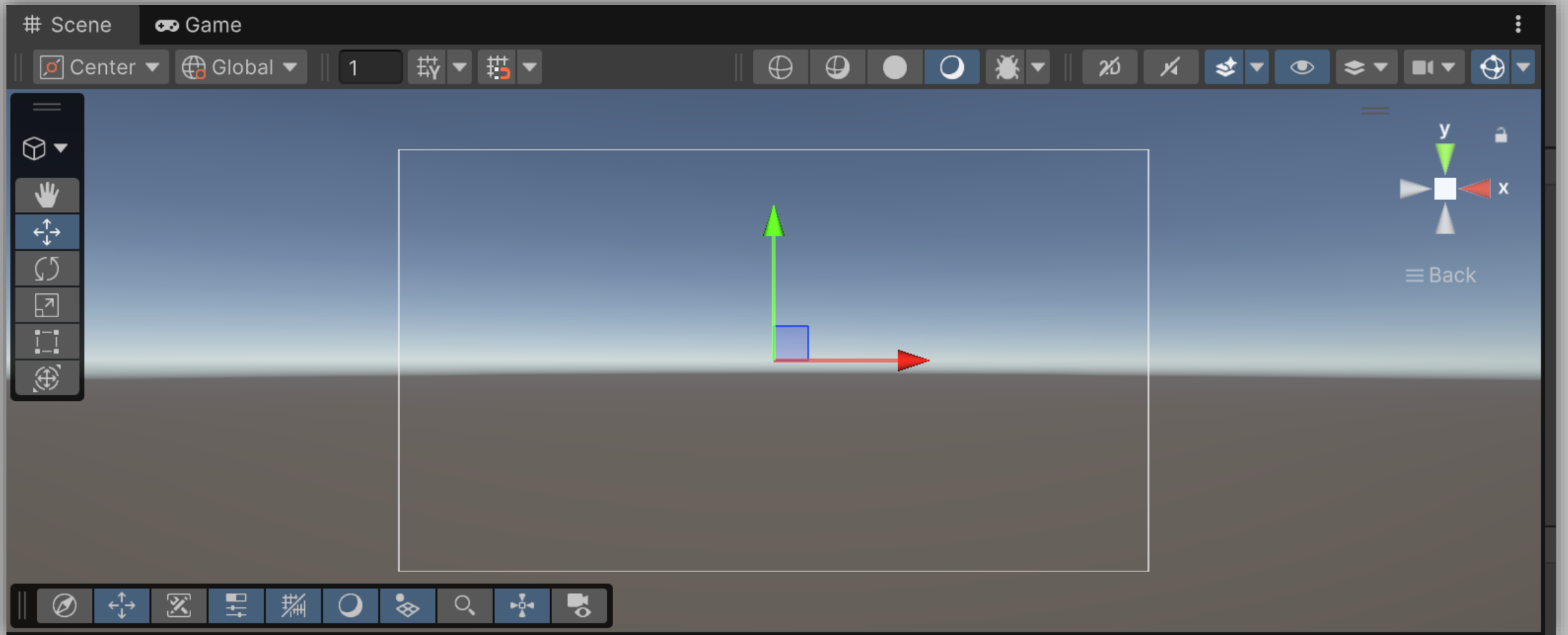
- Toggle
- Slider
- Scrollbar
- Scroll View
- Button - TextMeshPro
- Dropdown - TextMeshPro
- Input Field - TextMeshPro
- Canvas** (highlighted with a blue arrow)
- Event System

- Legacy >

Hierarchy panel showing the scene structure:

- miniGameNo1Scene*
 - Main Camera
 - Directional Light
 - PLayer1a**
 - Floor
 - Canvas
 - EventSystem





กรอบสี่เหลี่ยมเปรียบเสมือนหน้าจอแสดงผลเมื่อเกมทำงาน หน้าทีของเราคือวาง UI ในกรอบนี้



Hierarchy

+ All

- miniGameNo1Scene
 - Main Camera
 - Directional Light
 - Player1a
 - Floor
 - Canvas
 - EventSystem

Project Console

+ Favorites

- All Materials
- All Models

Paste Special

Rename

Duplicate Ctrl+D

Delete Del

Select All

Deselect All

Invert Selection

Select Children

Find References in Scene

Set as Default Parent

Prefab >

Create Empty Ctrl+Shift+N

Create Empty Parent Ctrl+Shift+G

2D Object >

3D Object >

Effects >

Light >

Audio >

Video >

UI >

1

Image

Text - TextMeshPro

Raw Image

Panel

Toggle

Slider

Scrollbar

Scroll View

Button - TextMeshPro

Dropdown - TextMeshPro

Input Field - TextMeshPro

Canvas

Event System


Legacy >



Hierarchy

All

- miniGameNo1Scene
 - Main Camera
 - Directional Light
 - Player1a**
 - Floor
 - Canvas
 - HPSlider**
 - EventSystem

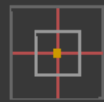



Inspector

HPSlider Static

Tag Untagged Layer UI

Rect Transform

| | | | |
|---|-------|--------|---|
| center | Pos X | Pos Y | Pos Z |
|  | 0 | 0 | 0 |
| middle | Width | Height | |
| | 160 | 20 |  R |

► Anchors

Pivot X 0.5 Y 0.5


Rotation X 0 Y 0 Z 0


Scale X 1 Y 1 Z 1


Slider


Interactable


Transition Color Tint


Target Graphic  Handle (Image)

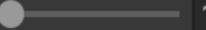
Normal Color 


Highlighted Color 

Pressed Color 

Selected Color 

Disabled Color 


Color Multiplier  1





Slider


Interactable


Transition Color Tint


Target Graphic  Handle (Image)

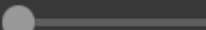
Normal Color 

Highlighted Color 

Pressed Color 

Selected Color 

Disabled Color 

Color Multiplier  1



Hierarchy

All

- miniGameNo1
 - Main Camera
 - Directional Light
 - Player1a
 - Floor
 - Canvas
 - HPSlider
 - EventSystem

Project Console

Favorites

- All Materials
- All Models

Paste Special

Rename

Duplicate Ctrl+D

Delete Del

Select All

Deselect All

Invert Selection

Select Children

Find References in Scene

Set as Default Parent

Prefab >

Create Empty Ctrl+Shift+N

Create Empty Parent Ctrl+Shift+G

2D Object >

3D Object >

Effects >

Light >

Audio >

Video >

UI >

1

Image

Text - TextMeshPro

Raw Image

Panel

Toggle

Slider

Scrollbar

Scroll View

Button - TextMeshPro

Dropdown - TextMeshPro

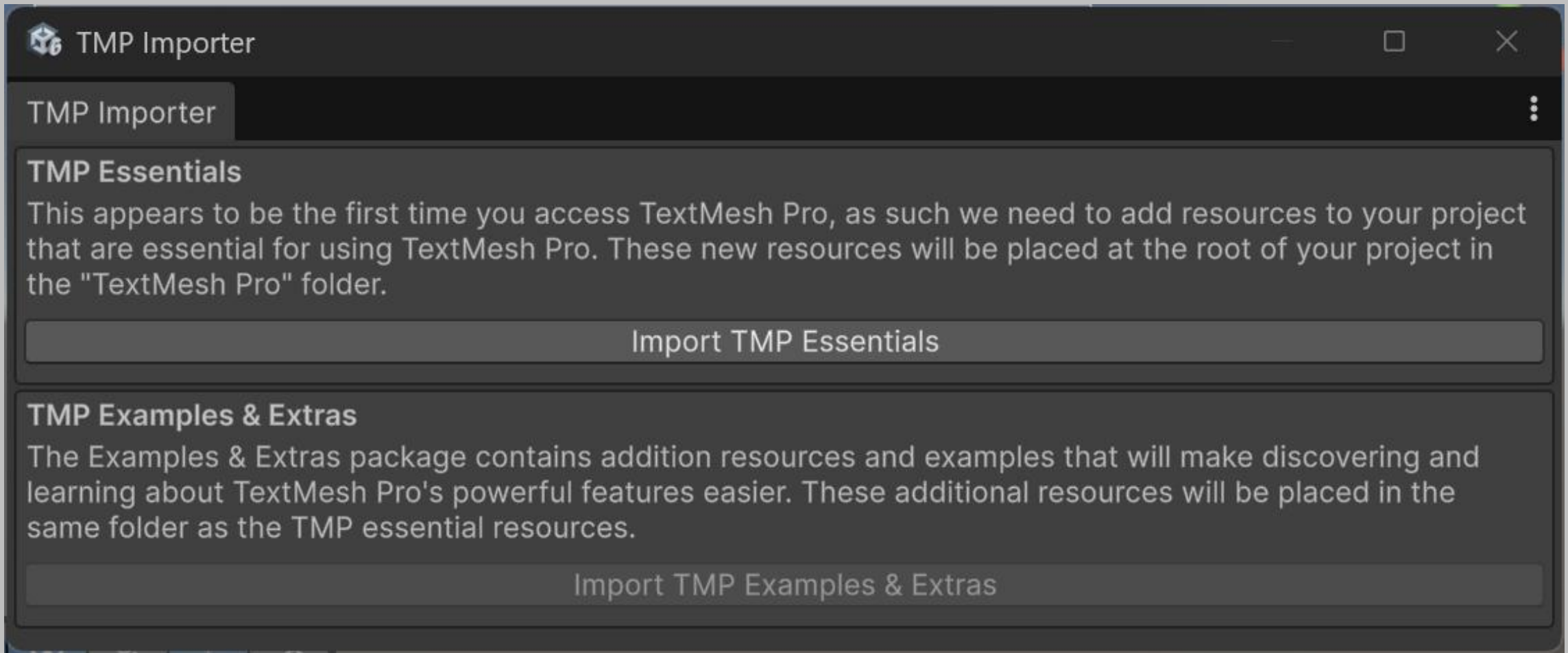
Input Field - TextMeshPro

Canvas

Event System

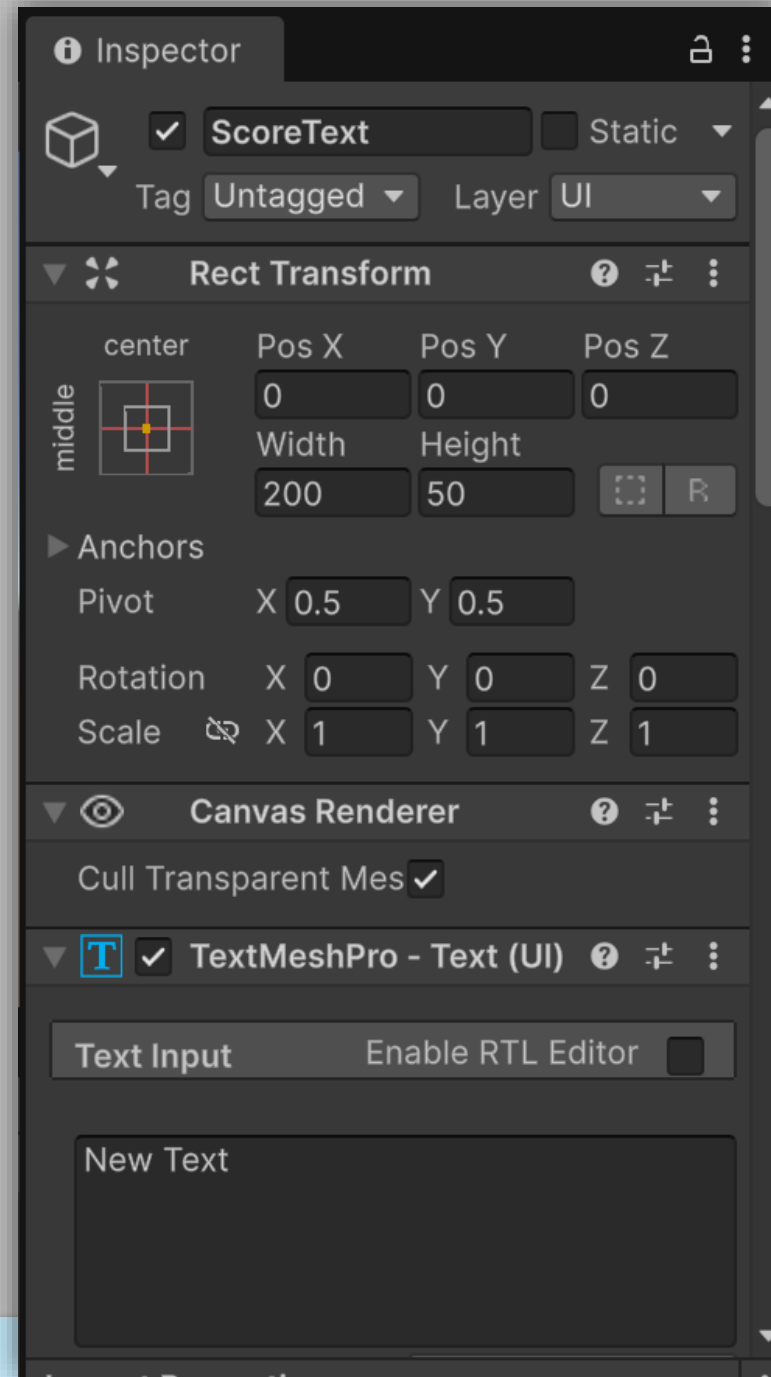
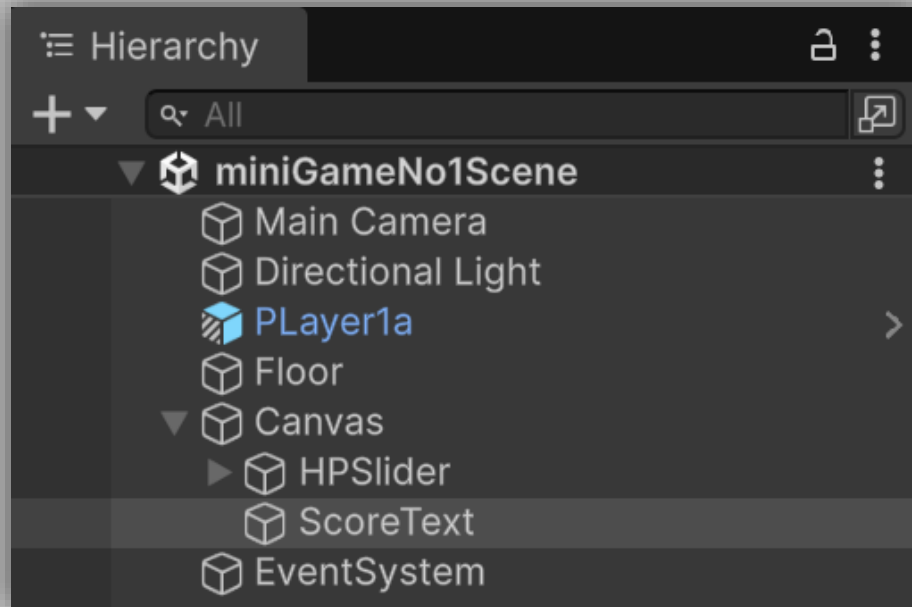
Legacy >

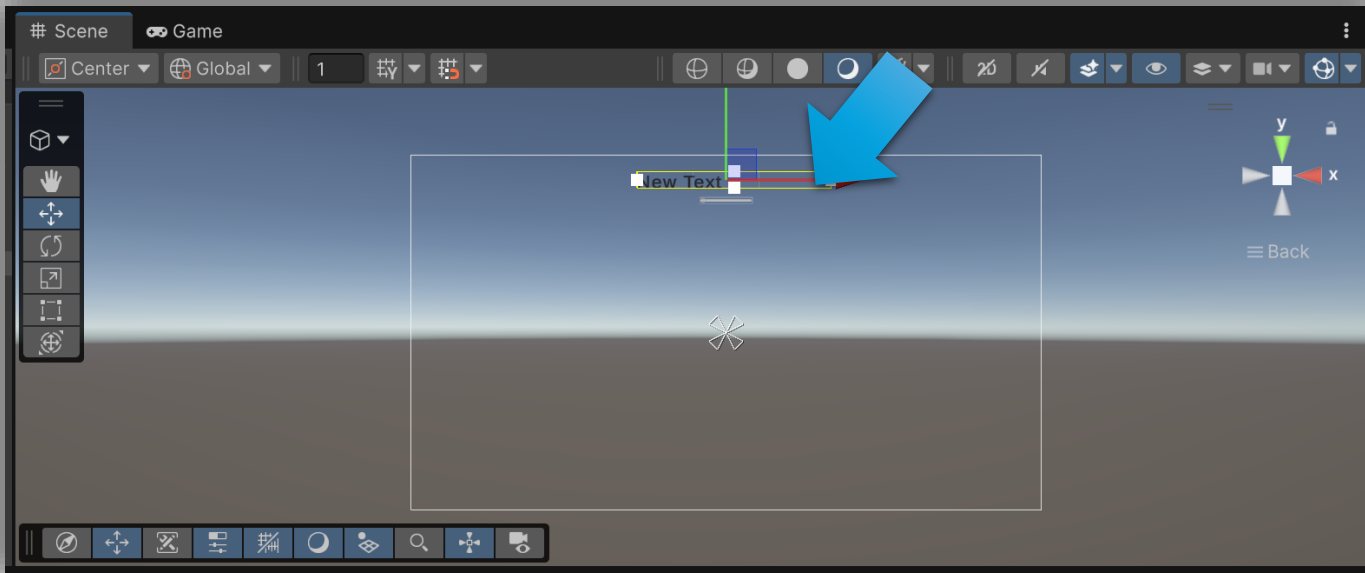




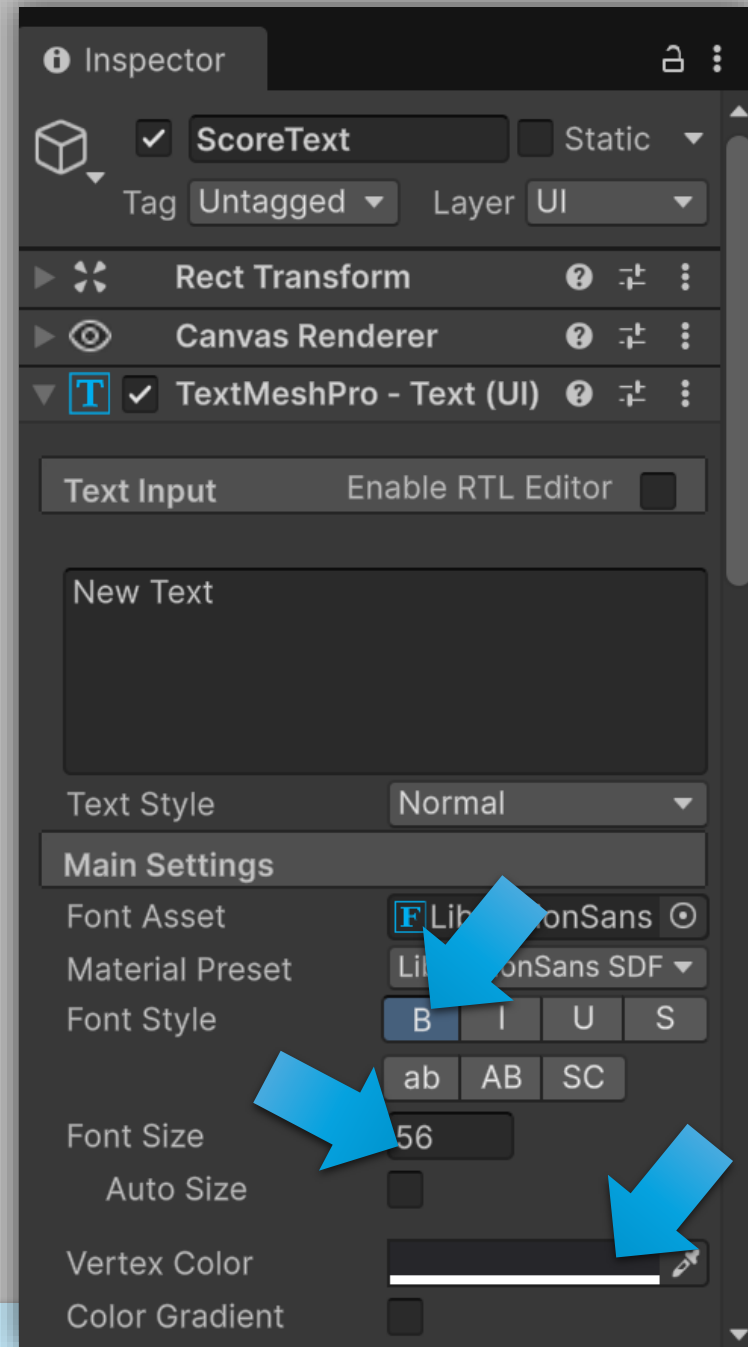
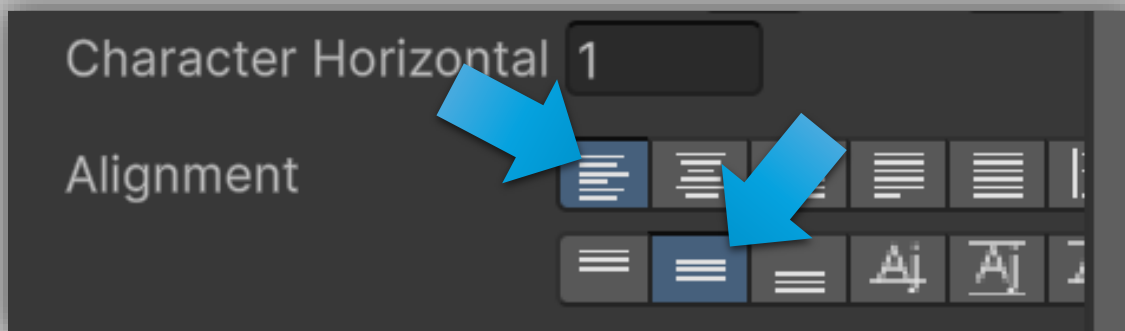
ในครั้งแรกที่มีการเพิ่ม Text – TextMesh Pro จะมีหน้าต่างถามเพื่อนำเข้าไลบรารี TMP ให้คลิกที่ปุ่ม Import ... หลังจากนั้นจะทำการติดตั้ง เมื่อติดตั้งเสร็จให้คลิกปุ่ม Import ... ปุ่มถัดไป เมื่อเสร็จสิ้นให้ปิดหน้าต่าง

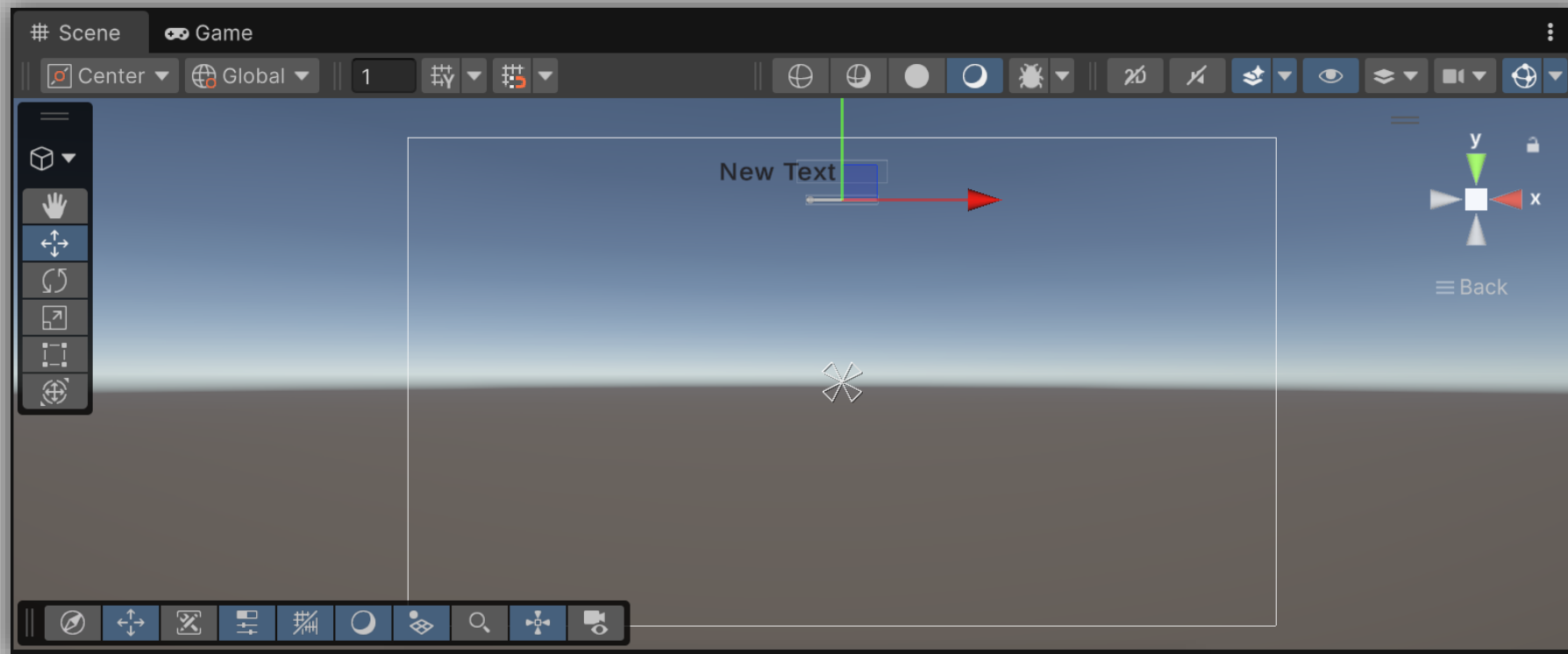






คลิกเลือก ScoreText เพื่อปรับลักษณะของ
ฟอนต์และจัดเรียงให้อยู่ตรงกลาง พร้อมทั้งปรับ
ขนาดของ ScoreText จากหน้าจอ Scene





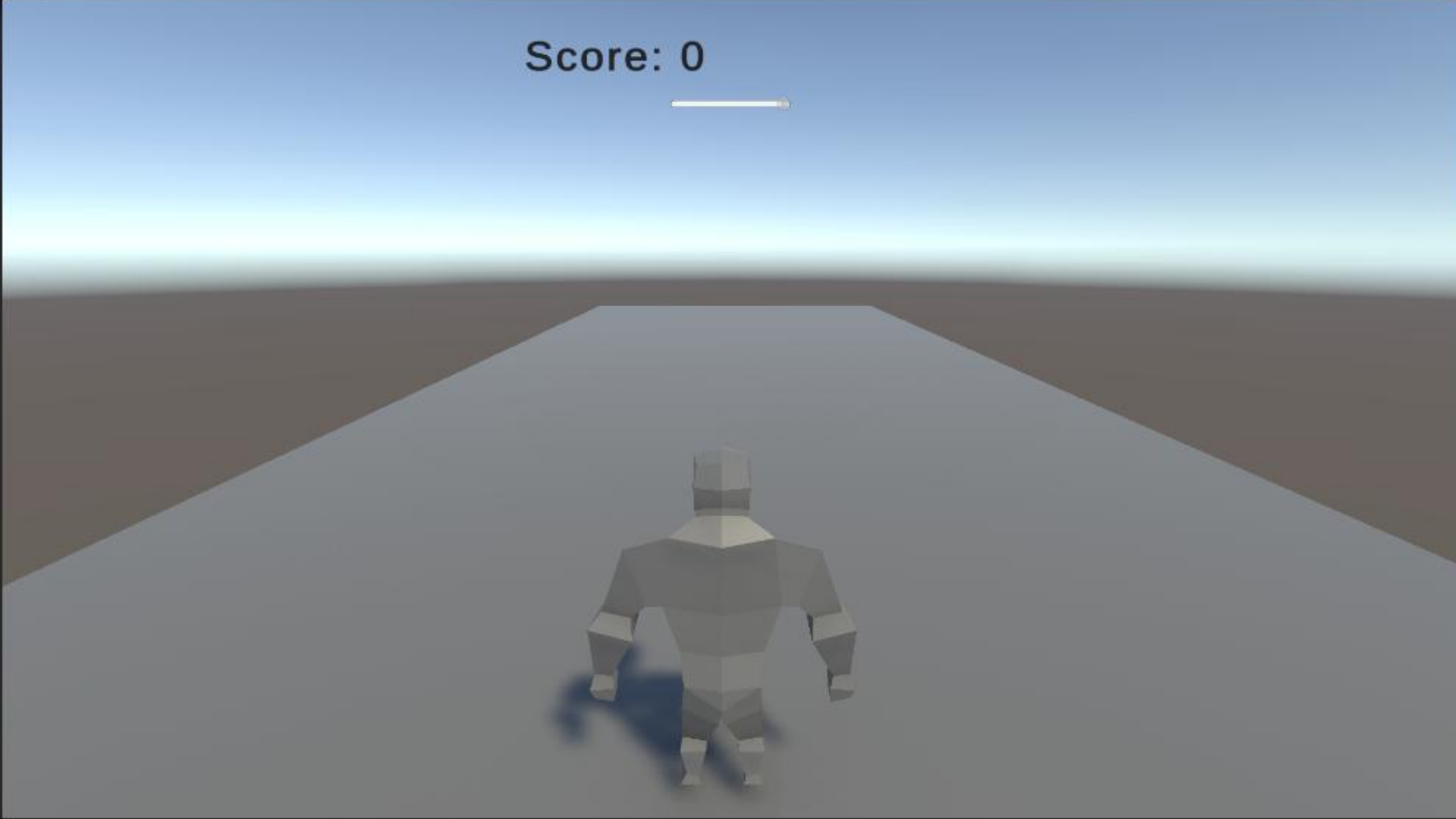
ย้ายตำแหน่ง HPSlider ไปไว้
ด้านล่างของ ScoreText



Game

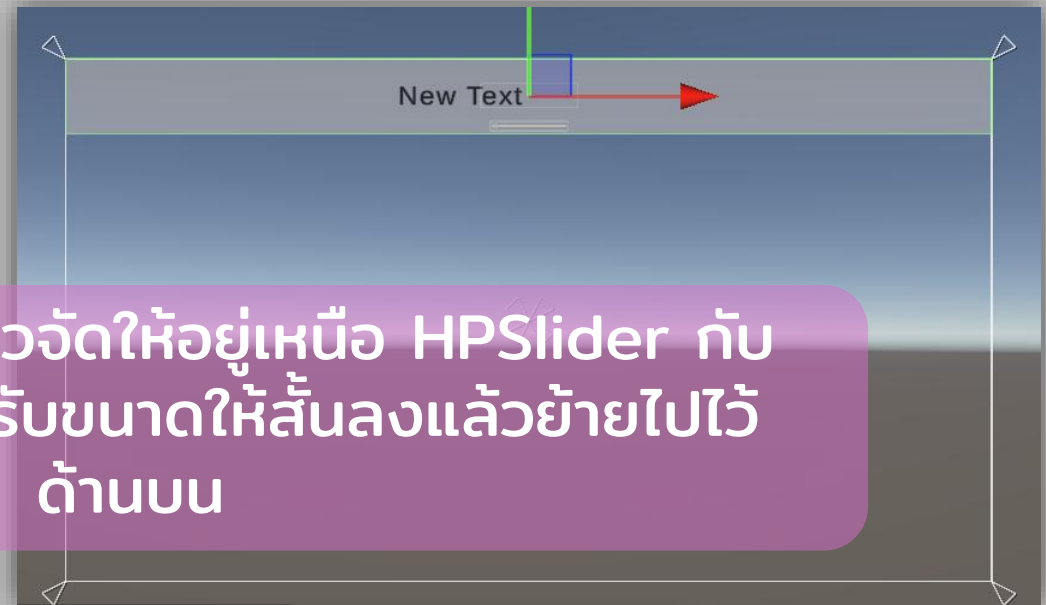
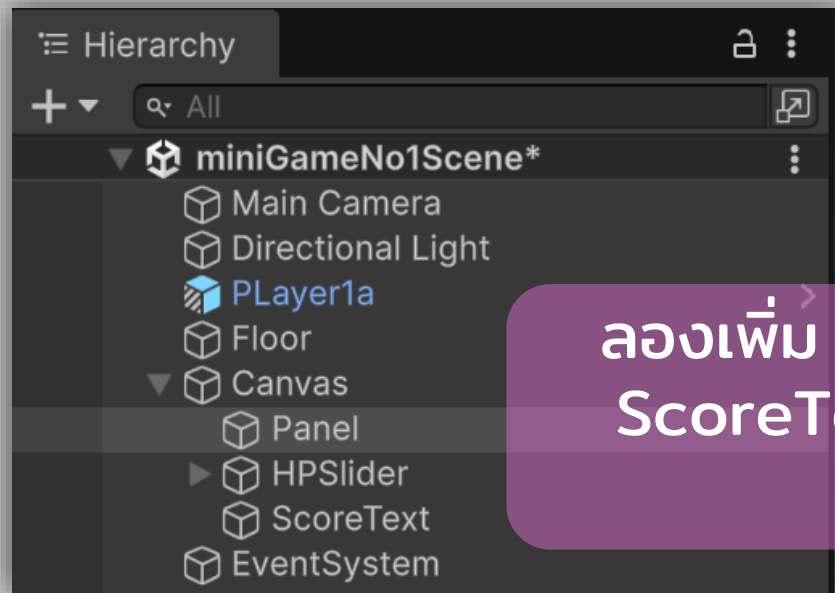
Display 1 Full HD (1920x1080) Scale 0.52x Play Focused Stats

Score: 0

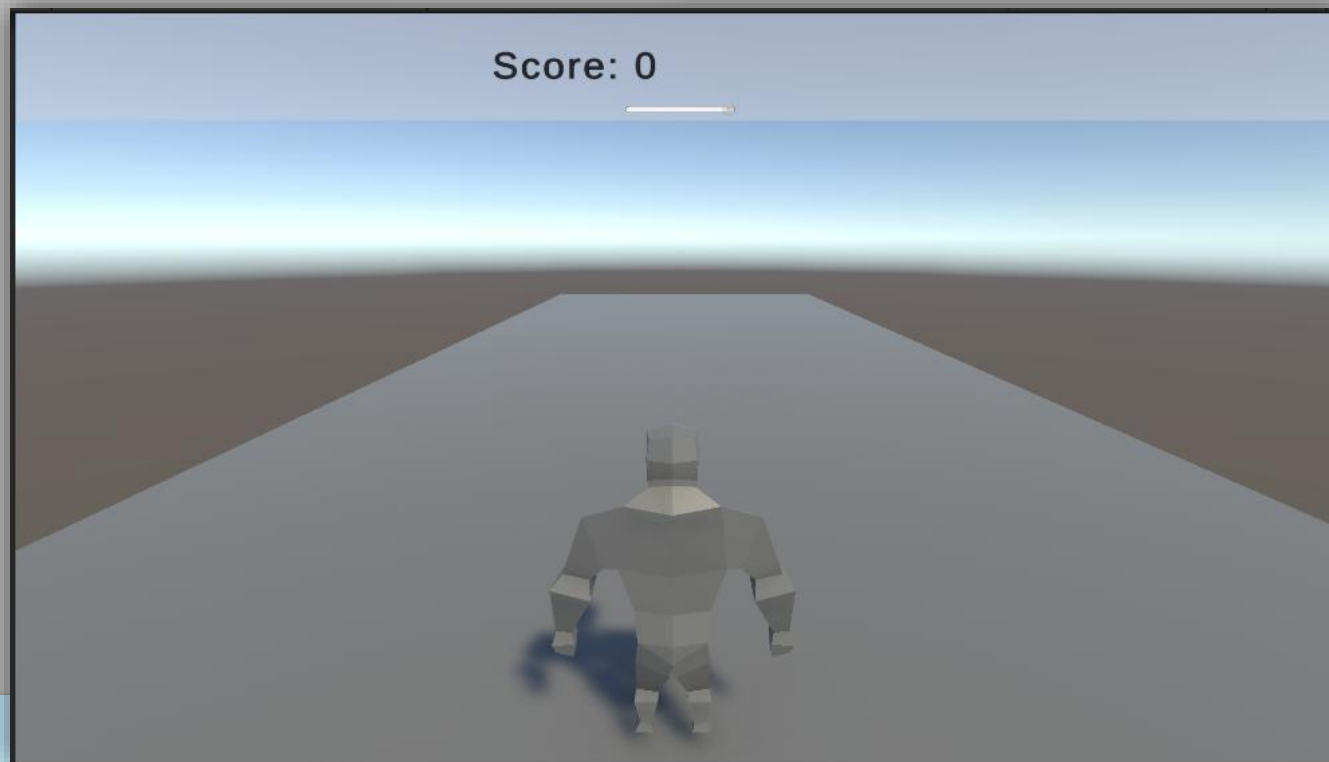


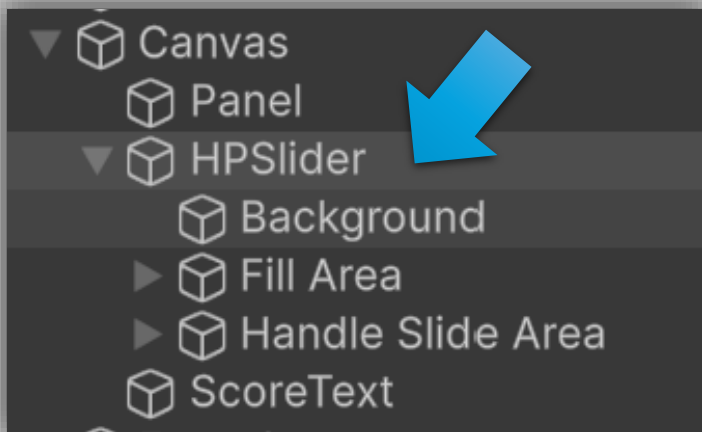
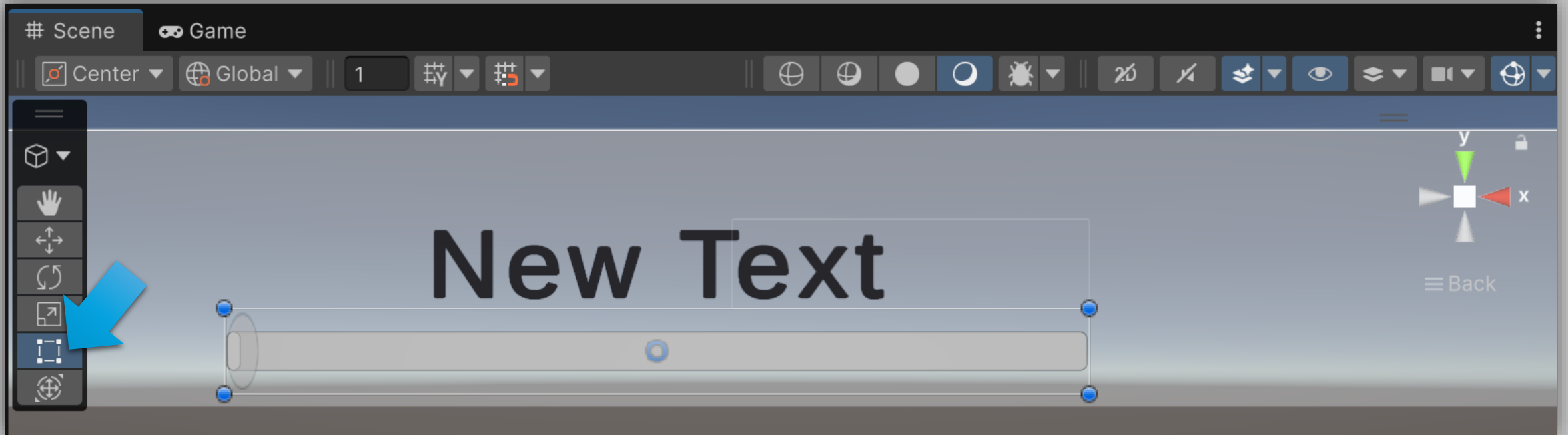
The image shows a game window with a dark interface. At the top, there's a 'Game' tab and a menu bar with options: 'Display 1', 'Full HD (1920x1080)', 'Scale' (set to 0.52x), 'Play Focused', and 'Stats'. The main game area displays a 3D scene with a low-poly character in the center, viewed from behind. The character is standing on a dark, flat ground. In the background, there's a bright, horizontal light beam or horizon line. The sky is a clear blue. At the top center of the game area, the text 'Score: 0' is displayed.





ลองเพิ่ม Panel แล้วจัดให้อยู่เหนือ HPSlider กับ ScoreText และปรับขนาดให้สั้นลงแล้วย้ายไปได้ด้านบน

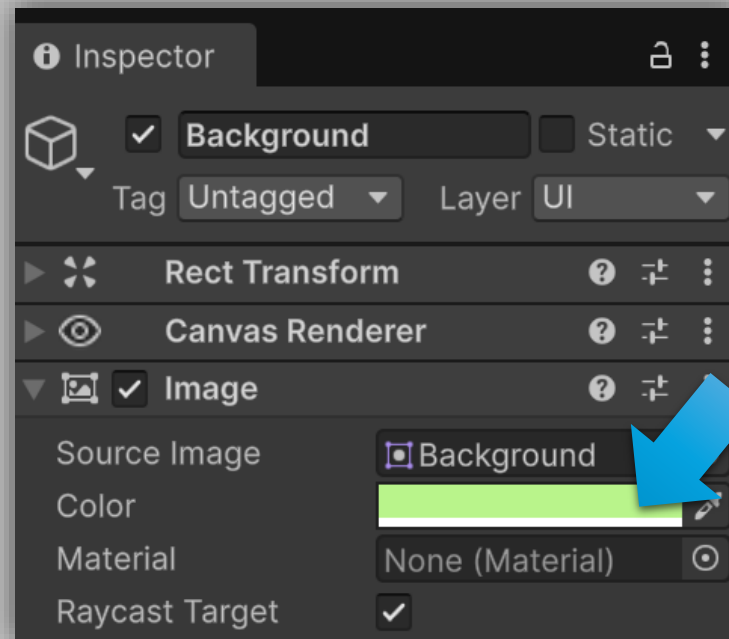
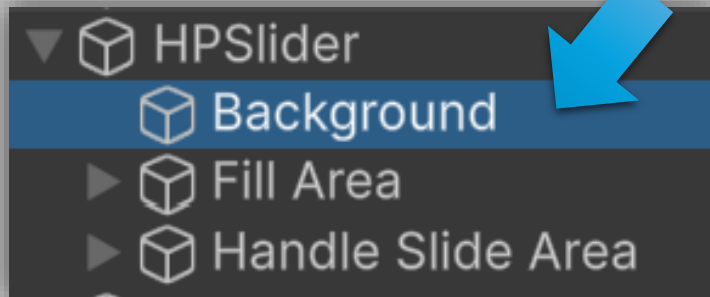




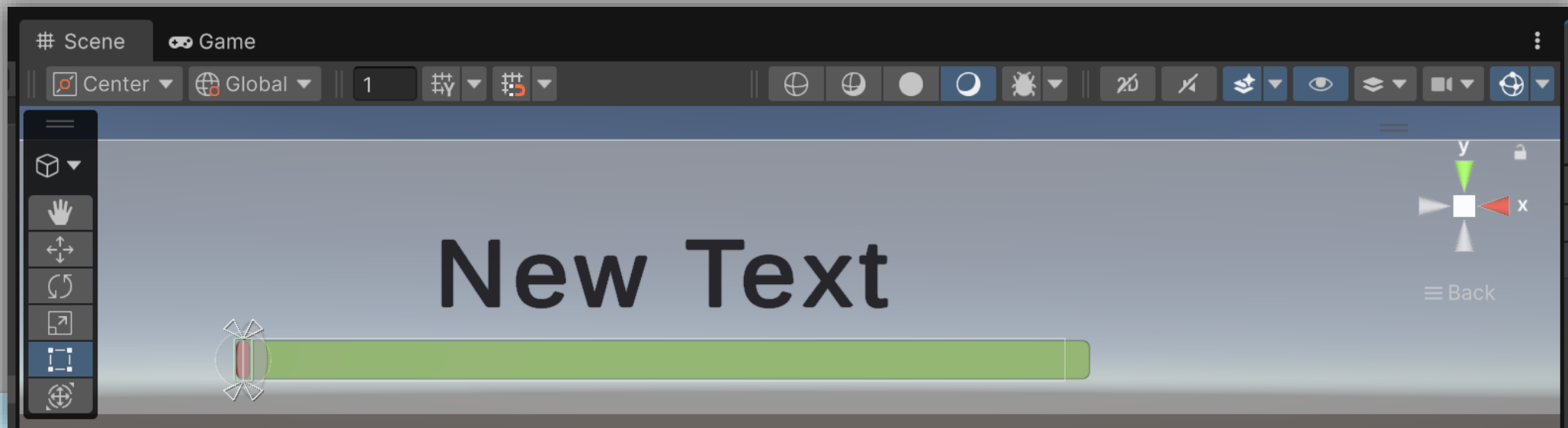
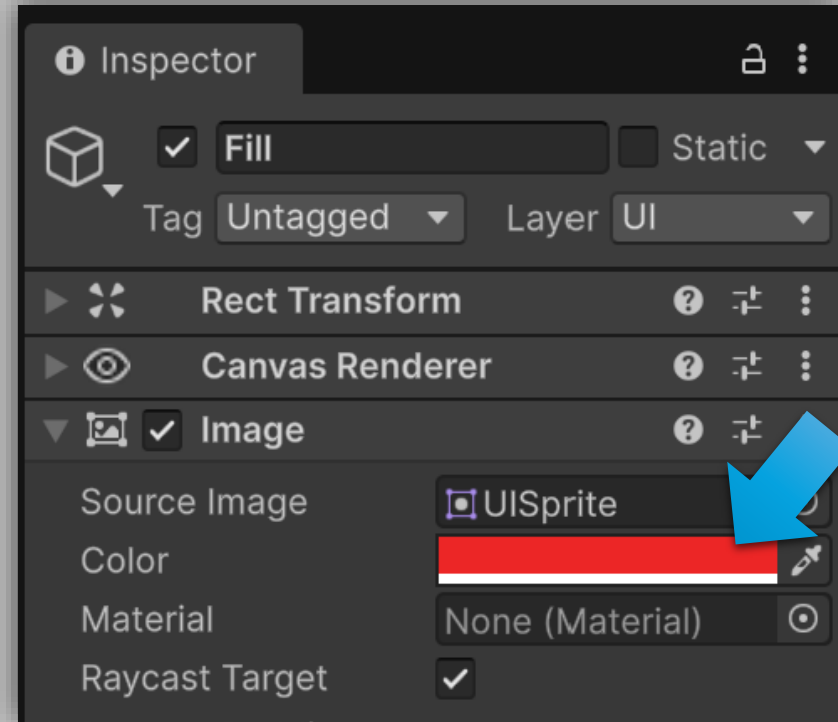
ปรับขนาดของ HPSlider



ปรับสีพื้นหลังของ HPSlider



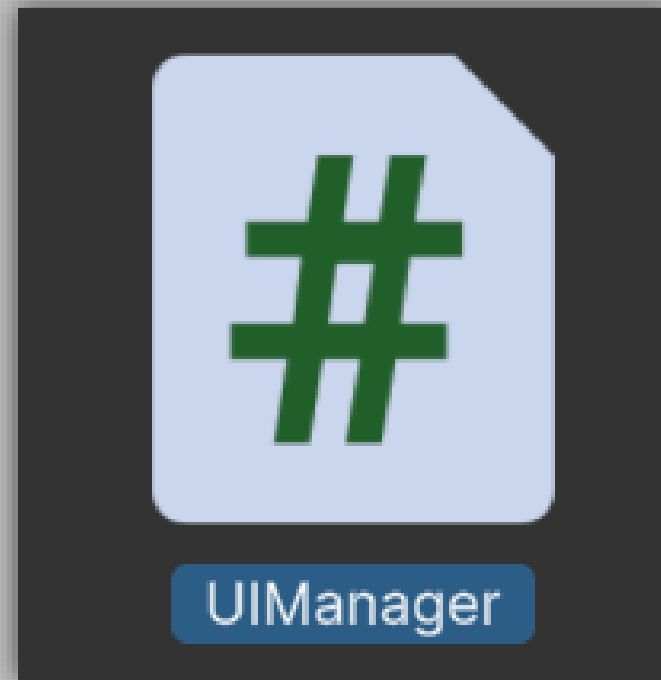
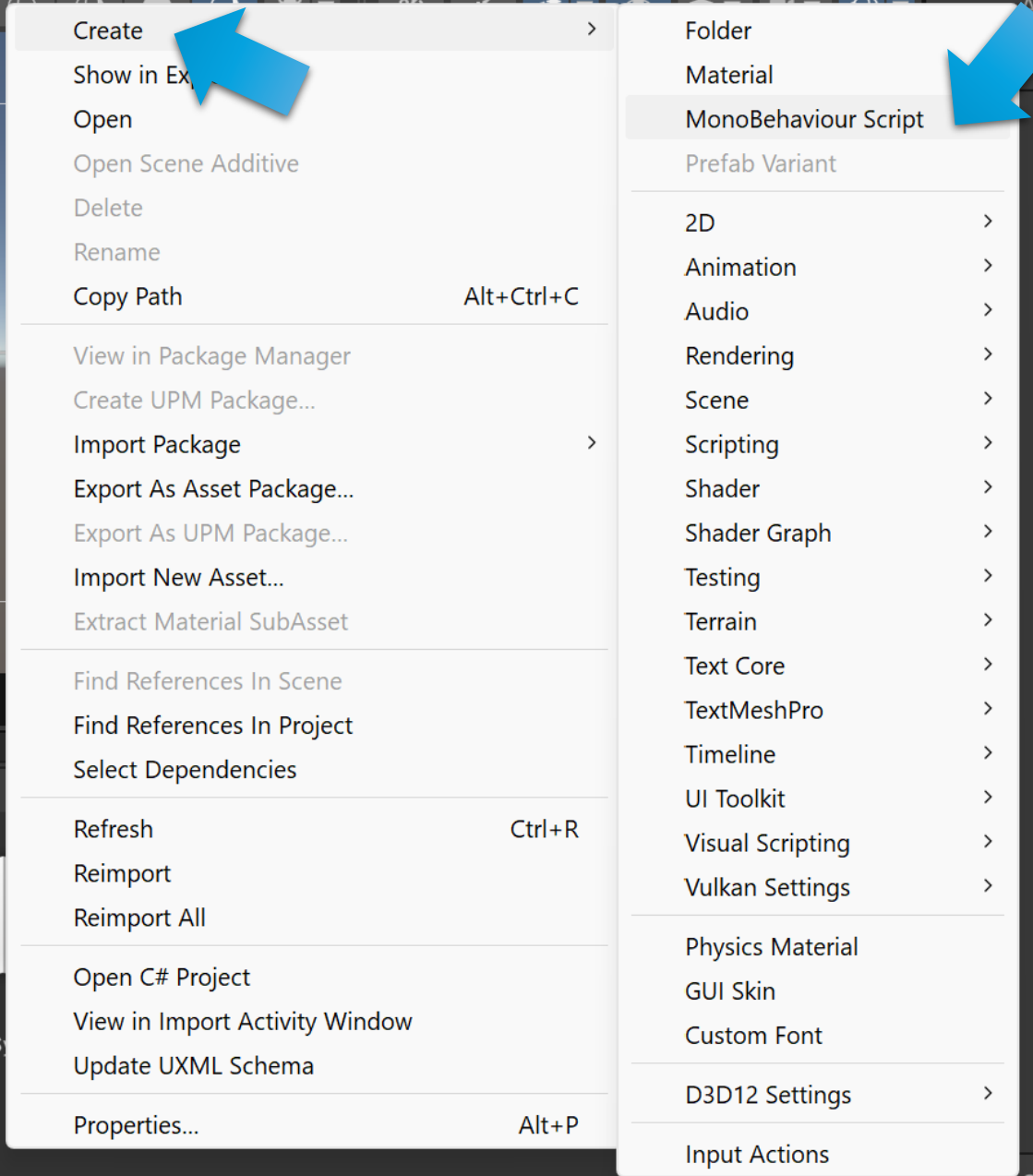
ปรับสีของแถบ HPSlider



การสร้าง UI Manager เพื่อแสดงผล

1. สร้าง Script แยกชื่อ UIManager.cs มาทำหน้าที่ "ดึง" ค่าจาก Player มาแสดงผลบนจอ
2. ลากไปใส่ใน Object ชื่อ EventSystem หรือสร้าง Empty Object ใหม่ชื่อ UIManager ก็ได้





UIManager.cs

PlayerActivity.cs

PlayerProperty.cs

Assembly-CSharp

UIManager

```
1  using TMPro;           // สำหรับ TMPro
2  using UnityEngine;
3  using UnityEngine.LowLevel;
4  using UnityEngine.UI; // สำหรับ Slider
5  public class UIManager : MonoBehaviour
6  {
7      [Header("UI References")]
8      public Slider hpSlider;
9      public TMProUGUI scoreText;
10     private GameObject player;
11     private PlayerProperty stats;
12
```



```
13 void Start()
14 {
15     // ค้นหา Object ที่มี Tag ว่า Player
16     player = GameObject.FindGameObjectWithTag("Player");
17     if (player != null)
18     {
19         stats = player.GetComponent<PlayerProperty>();
20     }
21 }
22
```

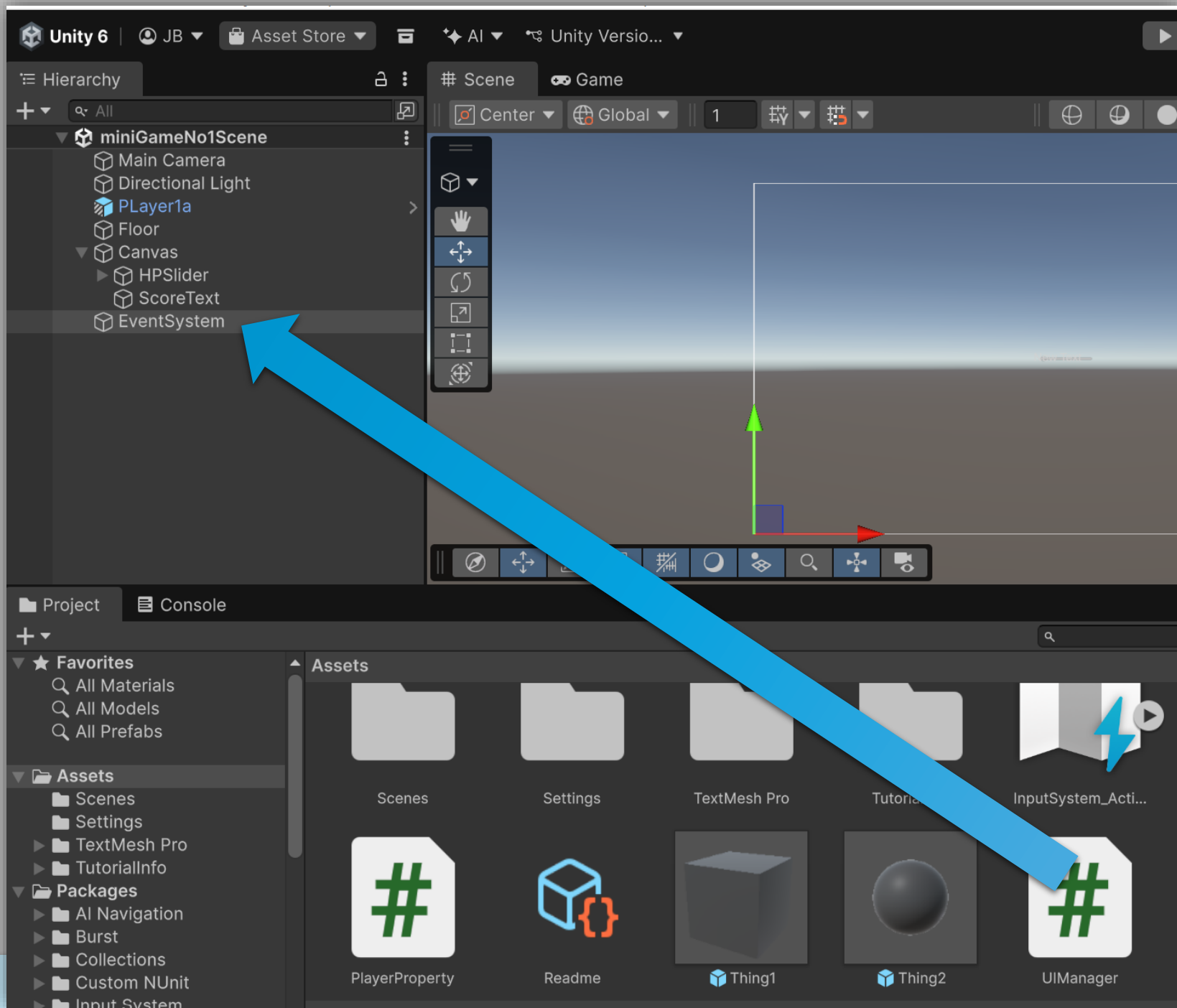


```
23  void Update()
24  {
25      // ถ้าพบ stats
26      if (stats != null)
27      {
28          UpdateUI();
29      }
30  }
31
```



```
32 void UpdateUI()
33 {
34     // อัปเดต HP Slider (ค่า Slider จะอยู่ระหว่าง 0 ถึง 1)
35     // ใช้ทศนิยมช่วยในการคำนวณ: $current / max$
36     hpSlider.value = (float)stats.hp / stats.hpMax;
37
38     // อัปเดต Score Text
39     scoreText.text = "Score: " + stats.score.ToString();
40 }
41 }
42
```





Inspector

EventSystem Static

Tag **Untagged** Layer **Default**

Transform

Event System

Script EventSystem

First Selected None (Game Object)

Send Navigation Events

Drag Threshold

Input System UI Input Module

UI Manager (Script)

Script UIManager

UI References

Hp Slider None (Slider)

Score Text None (Text Mesh Pi)

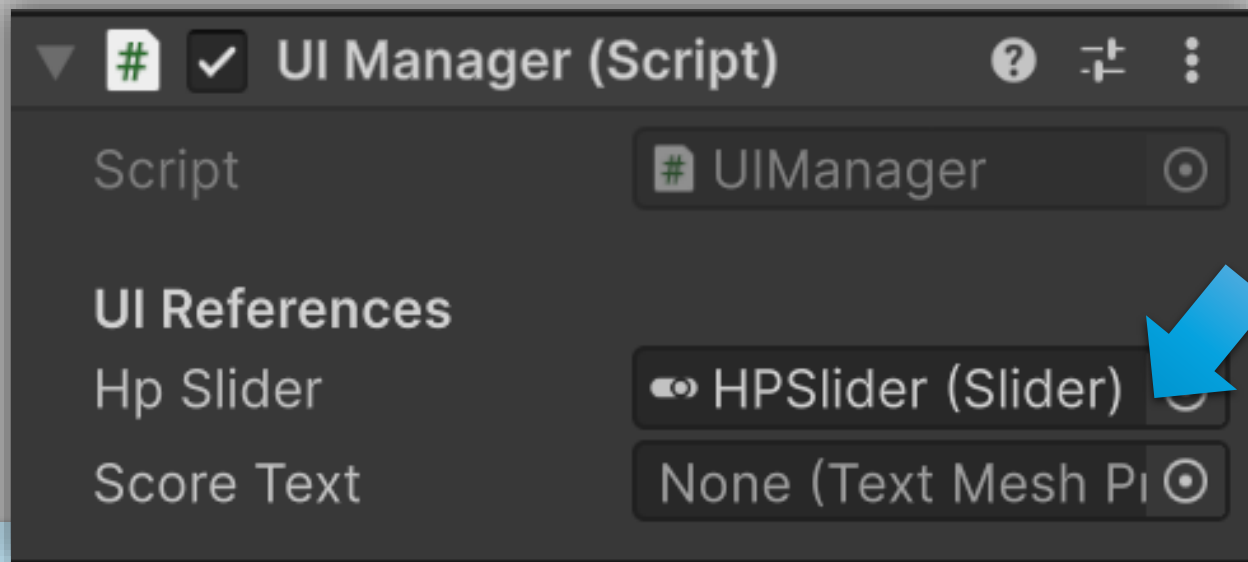
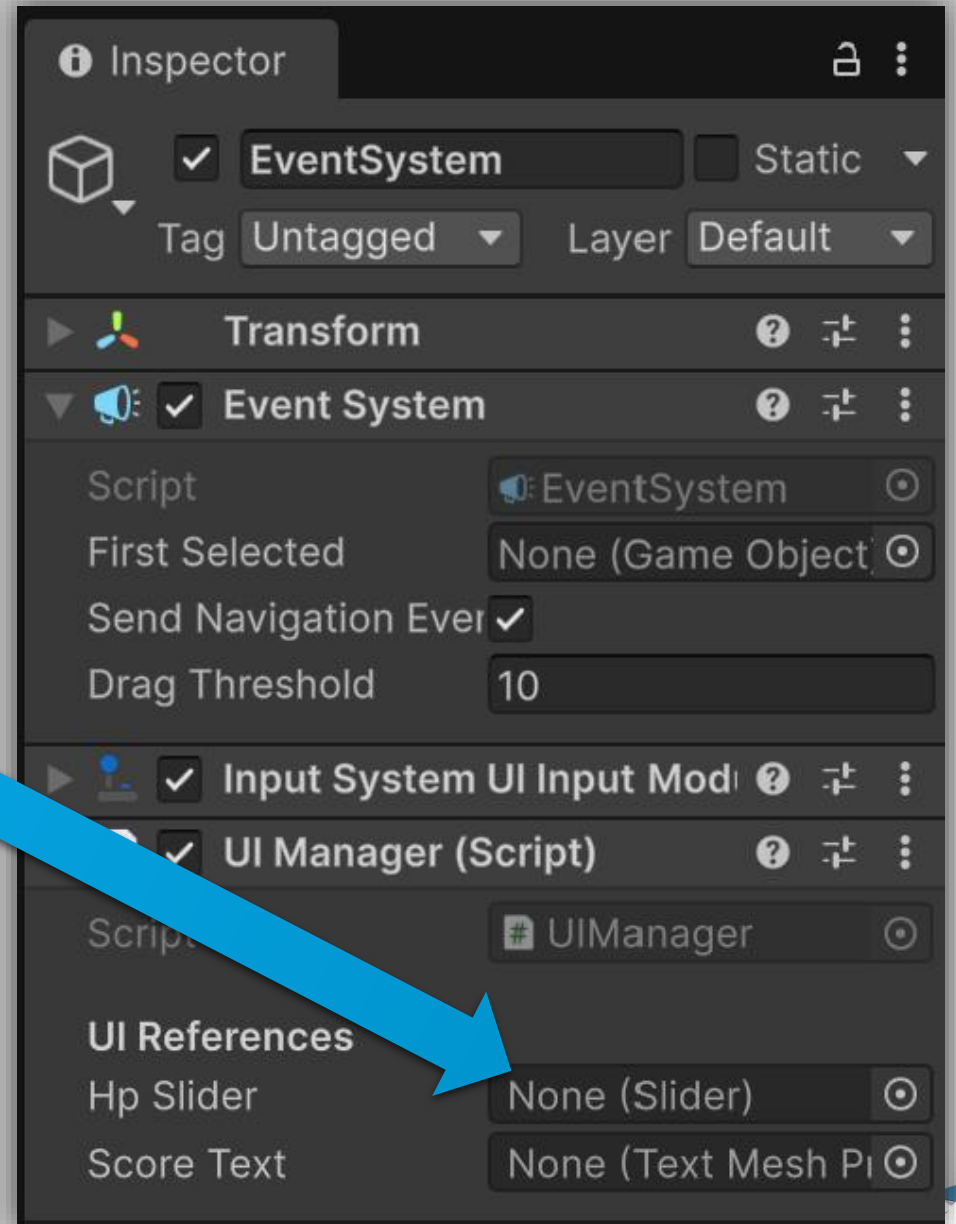
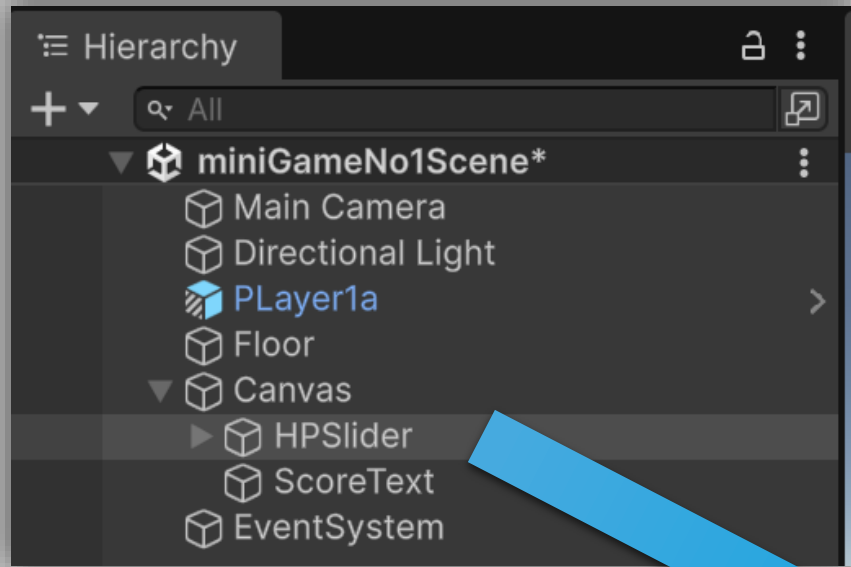


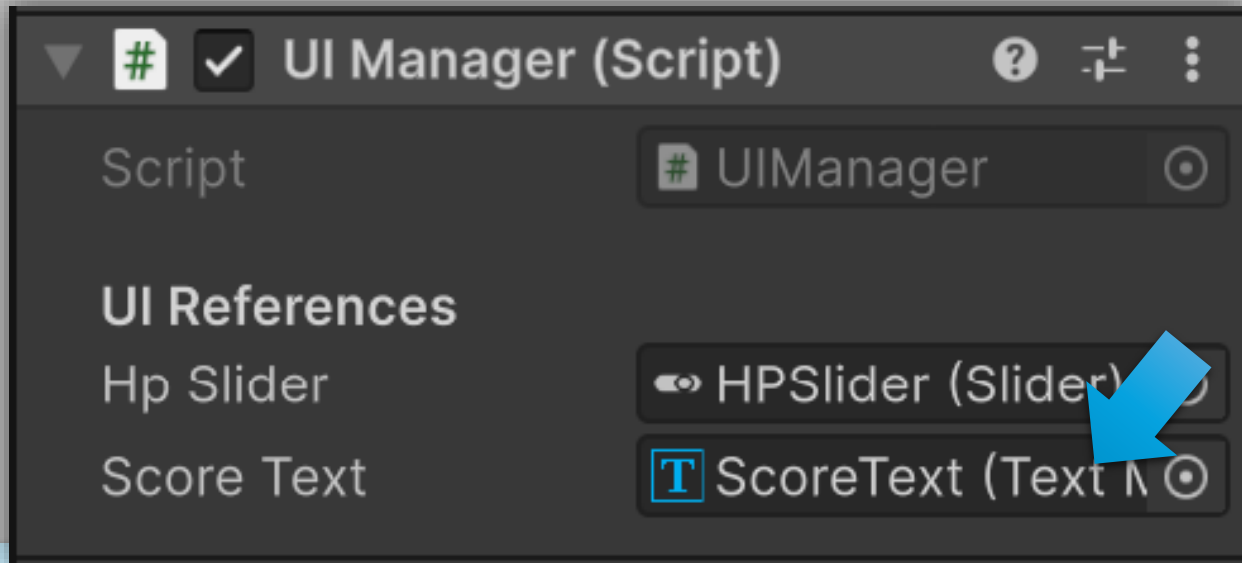
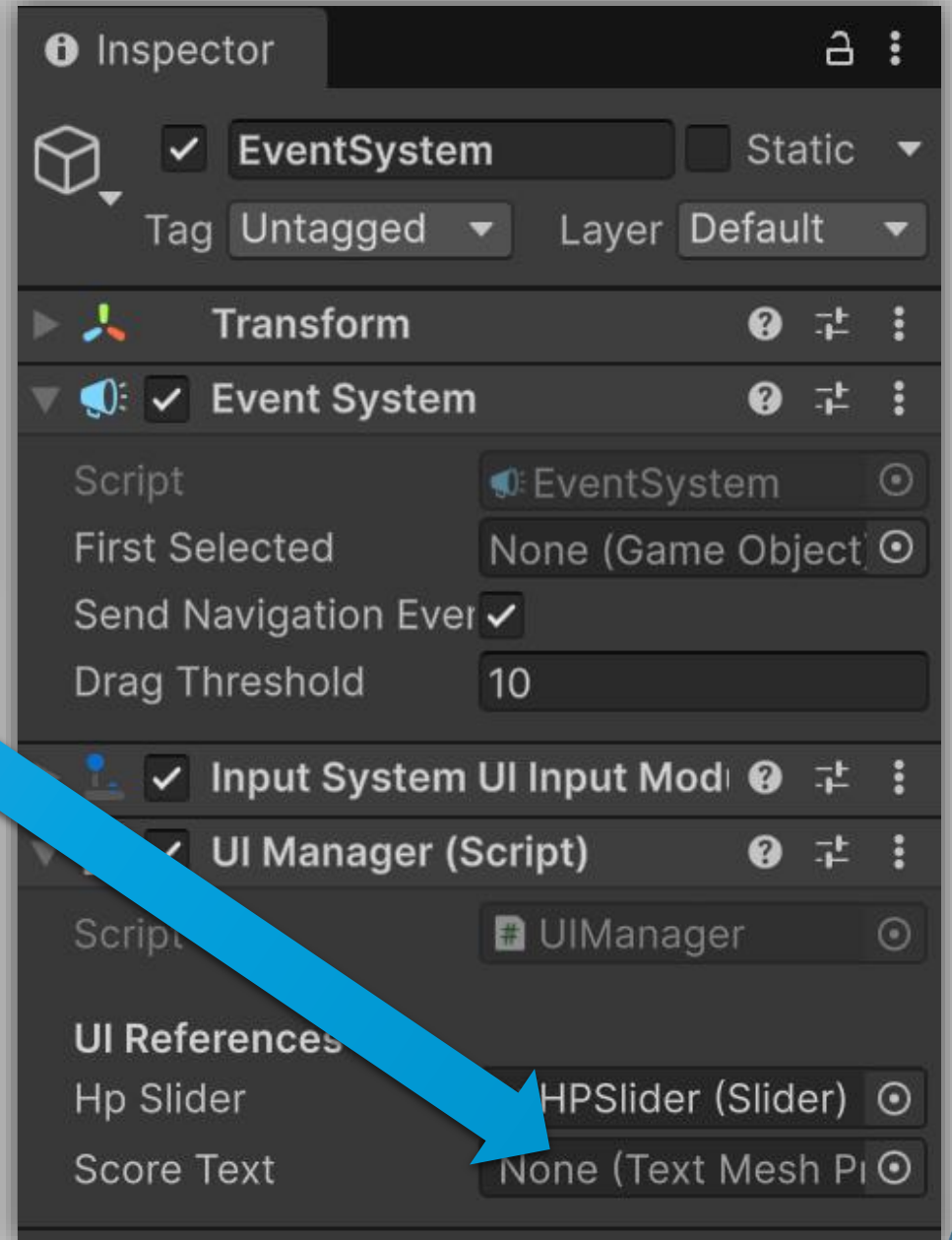
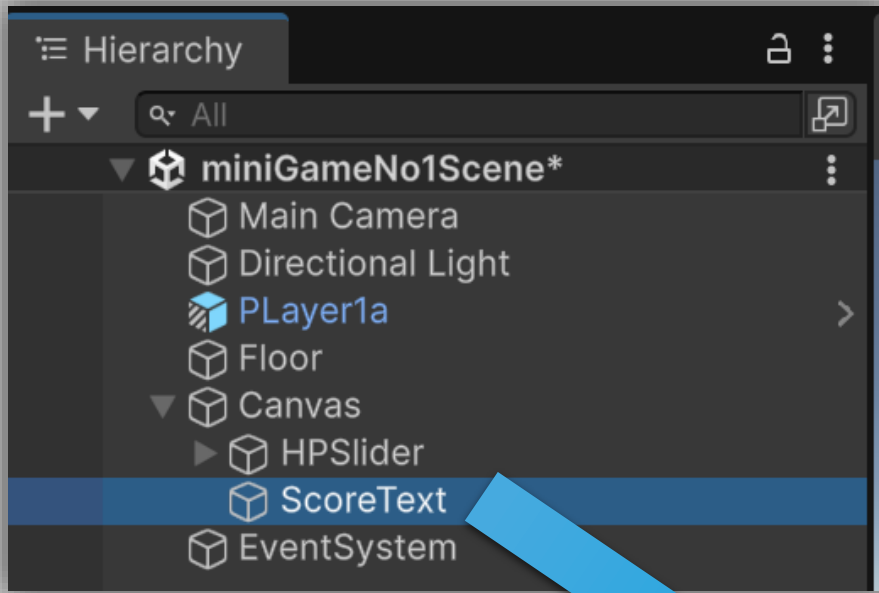
การเชื่อมต่อ (Wiring)

ขั้นตอนสุดท้ายคือการเชื่อมต่อ Object ในหน้า Inspector

1. คลิกที่ Object UIManager ใน Hierarchy
2. ลาก HPSlider จาก Hierarchy ไปใส่ในช่อง Hp Slider ใน Inspector
3. ลาก ScoreText จาก Hierarchy ไปใส่ในช่อง Score Text ใน Inspector







Performance

ใน Unity 6 การใช้ Update() เพื่อรับ UI **ทุกเฟรมเป็นการใช้ทรัพยากรที่สิ้นเปลือง (ไม่ประหยัดทรัพยากร)** ดังนั้น ถ้าเกมขนาดใหญ่ ควรเลือกใช้ C# Events หรือ Unity Events เพื่อส่งอัปเดต UI เฉพาะตอนที่ค่า hp หรือ score เปลี่ยนแปลงเท่านั้น



การใช้ Unity Event เป็นวิธีช่วยลดภาระของ CPU ที่ต้องคอยเช็คค่าในฟังก์ชัน Update() ทุกเฟรม (ซึ่งอาจจะรับถึง 60-120 ครั้งต่อวินาทีโดยไม่จำเป็น)

จึงเปลี่ยนมาใช้ระบบ "Observer Pattern" คือ ให้ตัวละคร "ตะโกนบอก" เมื่อมีการเปลี่ยนแปลงค่า แล้วให้ UI "รอฟัง" เพื่ออัปเดต



```
1  using UnityEngine;
2  using UnityEngine.Events;
3  using UnityEngine.InputSystem;
4  public class PlayerActivity : MonoBehaviour
5  {
6      private GameObject player;
7      private PlayerProperty stats;
8
9      // สร้าง Event สำหรับส่งสัญญาณ (สามารถลากใส่ใน Inspector ได้)
10     public UnityEvent onStatsChanged;
11     void Start()
12     {
13         // สั่งให้อัปเดท UI ครั้งแรกสุด
14         onStatsChanged?.Invoke();
15
16         player = GameObject.FindWithTag("Player");
17         if (player != null)
18         {
19             stats = player.GetComponent<PlayerProperty>();
20         }
21     }
```



```
22
23 public float speed = 5.0f;
    Unity Message | 0 references
24 void Update()
25 {
26     if (Keyboard.current.aKey.isPressed)
27     {
28         MoveLeft();
29     }
30     if (Keyboard.current.dKey.isPressed)
31     {
32         MoveRight();
33     }
34 }
35
```



```
36     void MoveLeft()
37     {
38         Vector3 curPos = transform.position;
39         Vector3 direction = Vector3.left * speed * Time.deltaTime;
40         curPos += direction;
41         if (curPos.x > -1.29)
42         {
43             transform.Translate(direction);
44         }
45     }
```



```
46 void MoveRight()
47 {
48     Vector3 curPos = transform.position;
49     Vector3 direction = Vector3.right * speed * Time.deltaTime;
50     curPos += direction;
51     if (curPos.x < 1.29)
52     {
53         transform.Translate(direction);
54     }
55 }
```



```
56  ✓
57  |
58  |
59  |
60  |
61  |

void IncScore()
{
    stats.score += 1;
    // "ตะโกนบอก" ว่าข้อมูลเปลี่ยนแล้วนะ!
    onStatsChanged?.Invoke();
}
```



```
62     void DecScore()  
63     {  
64         stats.score -= 1;  
65         if (stats.score < 0)  
66         {  
67             stats.hp -= 1;  
68             stats.score = 0;  
69         }  
70         onStatsChanged?.Invoke();  
71     }  
72 }
```



ปรับปรุง UIManager.cs

ในฟังก์ชัน UI จะไม่ใช้ Update() อีกต่อไป แต่จะสร้างฟังก์ชันที่รอรับการเรียกใช้จาก Player แทน



```
UIManager.cs  PlayerActivity.cs  PlayerProperty.cs
Assembly-CSharp  UIManager
1  using TMPro; // สำหรับ TMPro
2  using UnityEngine;
3  using UnityEngine.LowLevel;
4  using UnityEngine.UI; // สำหรับ Slider
5  public class UIManager : MonoBehaviour
6  {
7      [Header("UI References")]
8      public Slider hpSlider;
9      public TMProUGUI scoreText;
10     private GameObject player;
11     private PlayerProperty stats;
12     private PlayerActivity acts;
13
```



```
14 void Start()
15 {
16     // ค้นหา Object ที่มี Tag ว่า Player
17     player = GameObject.FindGameObjectWithTag("Player");
18     if (player != null)
19     {
20         stats = player.GetComponent<PlayerProperty>();
21         acts = player.GetComponent<PlayerActivity>();
22         // สมัครรับข้อมูล (Subscribe) ผ่าน Code
23         // ทุกครั้งที่ onStatsChanged.Invoke() ถูกเรียก ฟังก์ชัน UpdateUI จะทำงานทันที
24         acts.onStatsChanged.AddListener(UpdateUI);
25     }
26 }
27
```






```
28  ✓
29  |
30  | // ถ้าพบ stats
31  ✓ | if (stats != null)
32  | | {
33  | |     UpdateUI();
34  | | }
35  | }
36  |
```



```
37 void UpdateUI()
38 {
39     if (stats == null)
40     {
41         return;
42     }
43
44     // อัปเดต HP Slider (ค่า Slider จะอยู่ระหว่าง 0 ถึง 1)
45     // ใช้ทศนิยมช่วยในการคำนวณ: $current / max$
46     hpSlider.value = (float)stats.hp / stats.hpMax;
47
48     // อัปเดต Score Text
49     scoreText.text = "Score: " + stats.score.ToString();
50 }
51
```



```
53    void OnDestroy()  
54      {  
55           if (acts != null)  
56          {  
57              acts.onStatsChanged.RemoveListener(UpdateUI);  
58          }  
59      }  
60  }
```



การใช้งานใน Unity Editor

1. หากมีปุ่มในเกมหรือศัตรูที่โจมตี
 1. กด q เรียก IncScore();
 2. กด e เรียก DecScore();
2. ข้อดีของ UnityEvent คือ ทำให้สามารถเพิ่ม "คนฟัง" ได้อีกมากมายใน Inspector โดยไม่ต้องเขียน Code เพิ่ม เช่น
 1. เมื่อคะแนนลด ให้กล้องสั่น (Cinemachine Impulse)
 2. เมื่อคะแนนลด ให้เล่นเสียง Effect
 3. เมื่อคะแนนลด ให้นำจอกระพริบสีแดง



Unity Message | 0 references

```
24     void Update()
25     {
26         if (Keyboard.current.aKey.isPressed)
27         {
28             MoveLeft();
29         }
30         if (Keyboard.current.dKey.isPressed)
31         {
32             MoveRight();
33         }
34         if (Keyboard.current.qKey.isPressed)
35         {
36             IncScore();
37         }
38         if (Keyboard.current.eKey.isPressed)
39         {
40             DecScore();
41         }
42     }
43
```

ปรับ Update() ของ
PlayerActivity.cs



```
64 public void IncScore(int damage=1)
65 {
66     stats.score += damage;
67     // "ตะโกนบอก" ว่าข้อมูลเปลี่ยนแล้วนะ!
68     onStatsChanged?.Invoke();
69 }
70 public void DecScore(int damage=1)
71 {
72     stats.score -= damage;
73     if (stats.score < 0)
74     {
75         stats.hp -= 1;
76         stats.score = 0;
77     }
78     onStatsChanged?.Invoke();
79 }
80 }
81
```

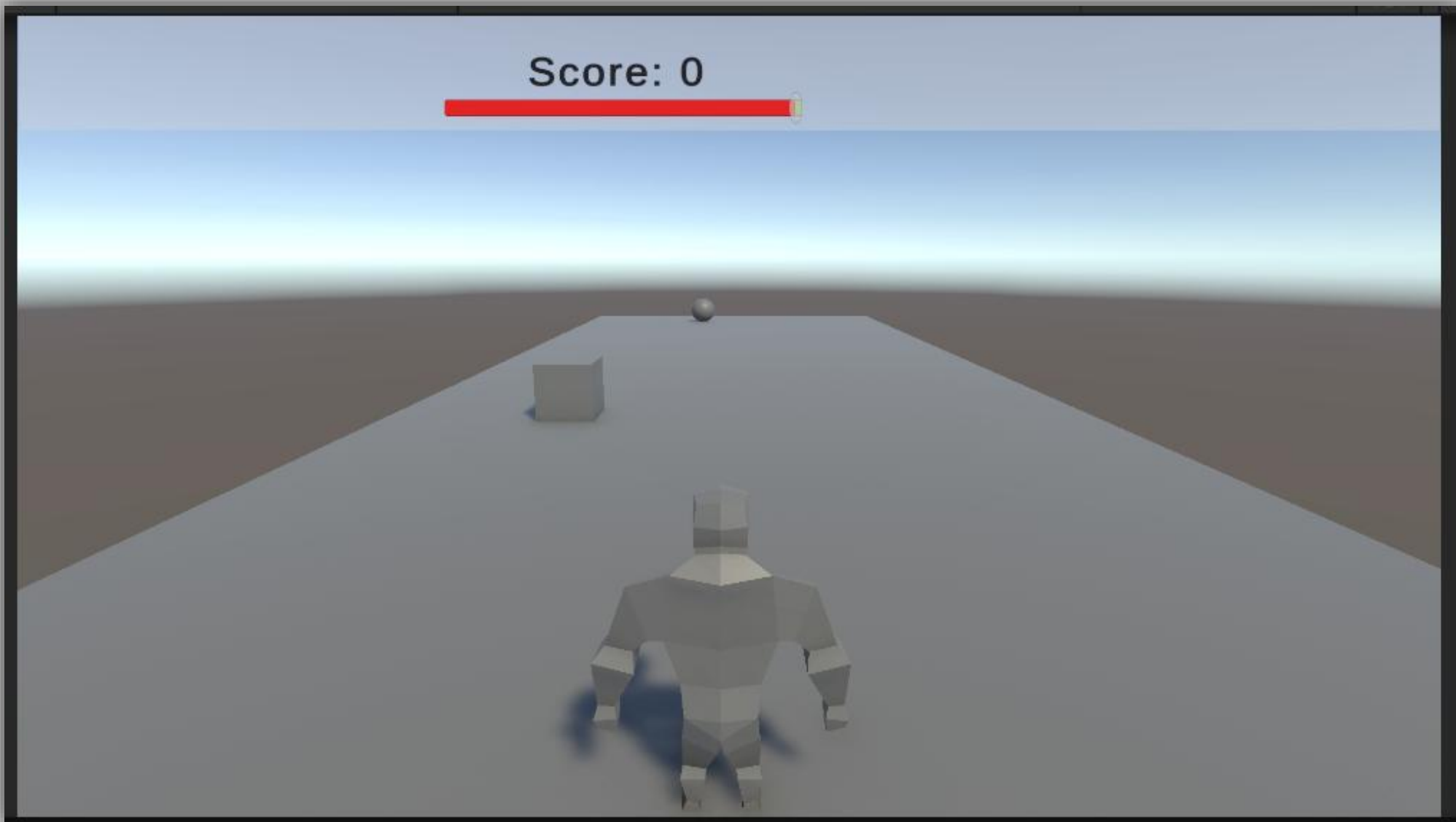
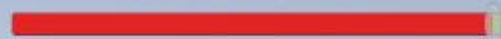
ปรับแก้
IncScore() และ
DecScore() ของ
PlayerActivity.cs



ลองเล่นเกมแล้วกด q หรือ e เพื่อเปลี่ยนค่า score และ
มีผลต่อ hp อีกด้วย (แต่ไวมาก ๆ)



Score: 0



Q & A

