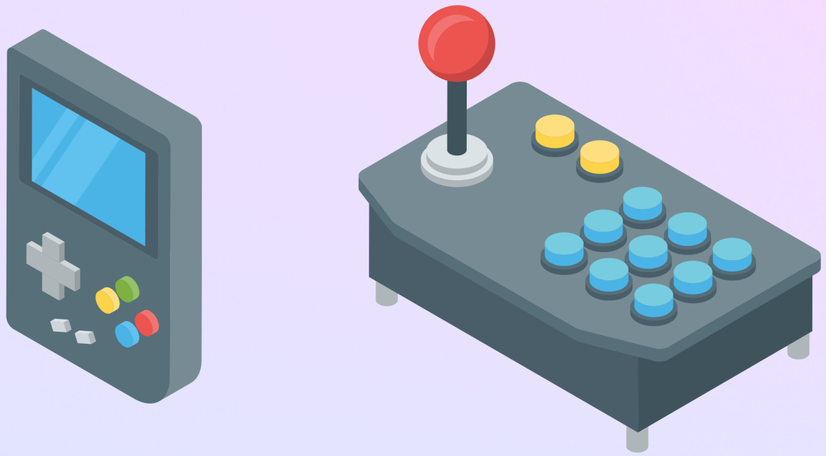


WOLVEDEN ACADEMY

NETWORKING AND MULTIPLAYER ONLINE GAMES

MULTIPLAYER NEW ERA

BY PONGSATHORN KIATTICHAROENPORN (MEW)

WEEK 5 : SAMPLE GAMEPLAY FOR MULTIPLAYER 2

Sample Gameplay for Multiplayer 2

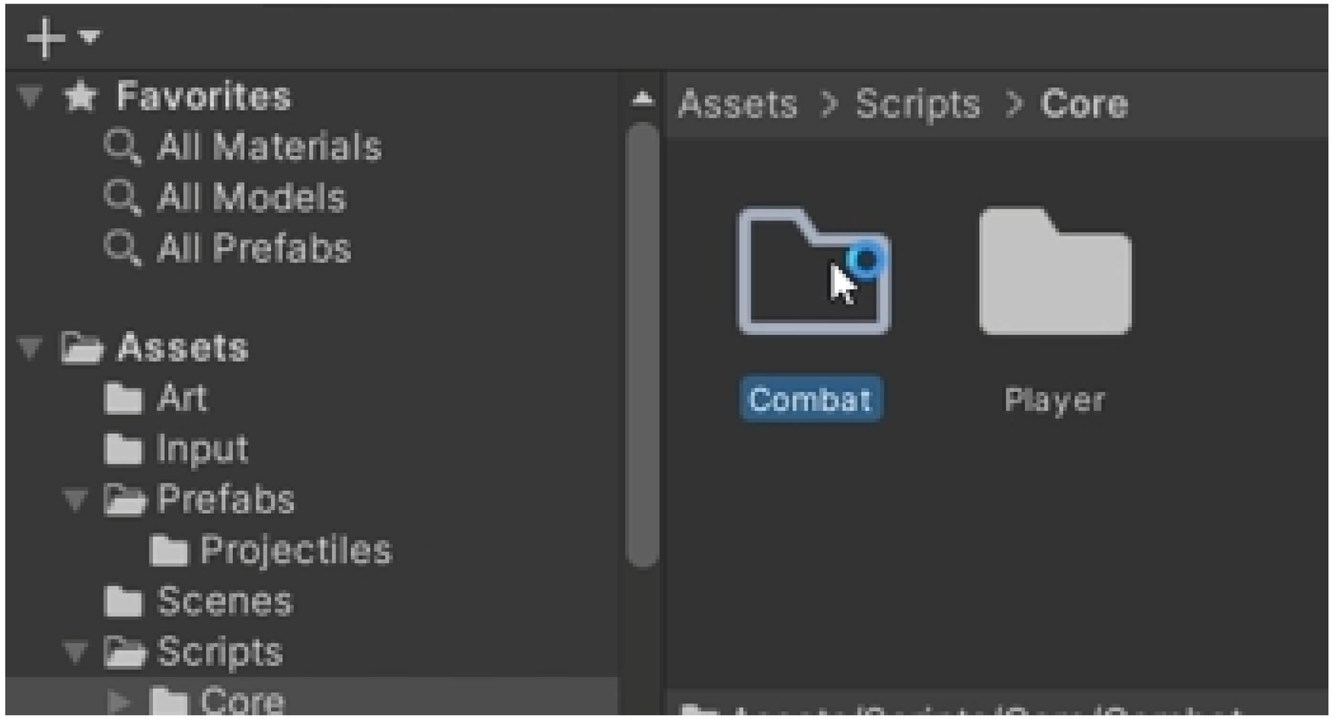
Topic in this week

Workshop

- Health Component
- Health Display
- Dealing Damage
- Coins
- Coin Wallet
- Coin Spawner
- Map Design

Health Component

สร้างโฟลเดอร์



สร้าง Script Health

```
using System.Collections;
using System.Collections.Generic;
using Unity.Netcode;
using UnityEngine;

public class Health : NetworkBehaviour
```

สิ่งนี้คือตัวแปรแบบ Network ที่จะ Sync กัน ระหว่าง Component ของตัว Object นั้นๆ

```
0 references
public NetworkVariable<int> CurrentHealth = new NetworkVariable<int>();
```

NetworkVariable | Unity Multiplayer Networking

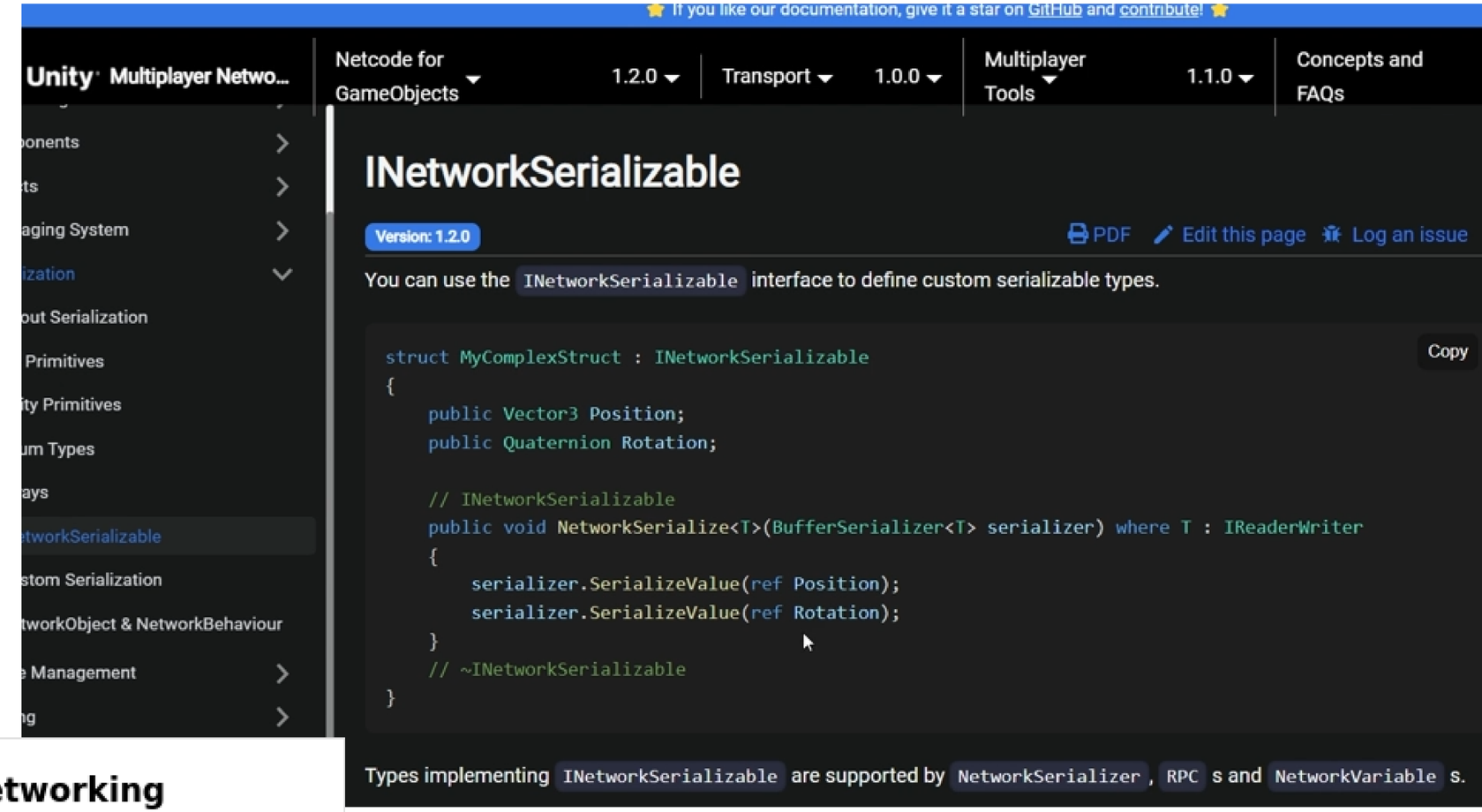
Introduction

unity3d.com

```
0 references
[field: SerializeField] public int MaxHealth { get; private set; } = 100;
```

เนื้อหา: รูปแบบการเก็บตัวแปรระหว่าง Network หากเป็นตัวแปร Basic จะส่งผ่านด้วยวิธีปกติได้เลย

แต่หากตัวแปรที่ซับซ้อน อย่าง Vector3 , Quaternion เป็นต้น ต้องมีการใช้วิธี INetworkSerializable มาช่วย



About Serialization | Unity Multiplayer Networking

Multiplayer framework has built-in serialization support for C# and Unity primitive types out-of-the-box, also with ability to further extend network serialization for user defined types implementing INetworkSerializable...

unity3d.com

```
0 references
public class Health : NetworkBehaviour
{
    1 reference
    [field: SerializeField] public int MaxHealth { get; private set; } = 100;

    1 reference
    public NetworkVariable<int> CurrentHealth = new NetworkVariable<int>();

    1 reference
    public override void OnNetworkSpawn()
    {
        if (!IsServer) { return; }

        CurrentHealth.Value = MaxHealth;
    }
}
```

```
0 references
public void TakeDamage(int damageValue)
{
    ModifyHealth(-damageValue);
}

0 references
public void RestoreHealth(int healValue)
{
    ModifyHealth(healValue);
}

2 references
private void ModifyHealth(int value)
{

}
```

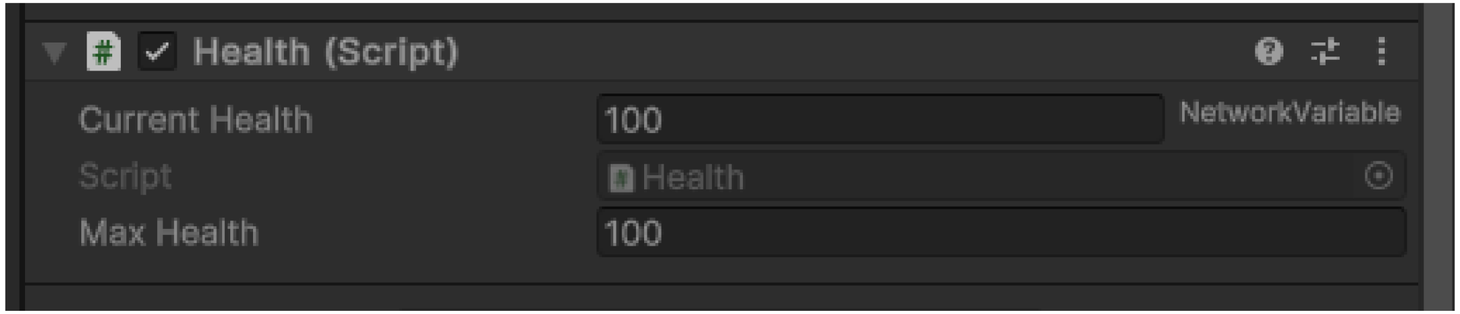
```
0 references
private bool isDead;

0 references
public Action<Health> OnDie;
```

Modify Health Method

- Make sure we aren't already dead
- Calculate new health value
- Clamp between 0 and **MaxHealth**
- If we have now reached 0 health, invoke the **OnDie** event and set the **isDead** bool

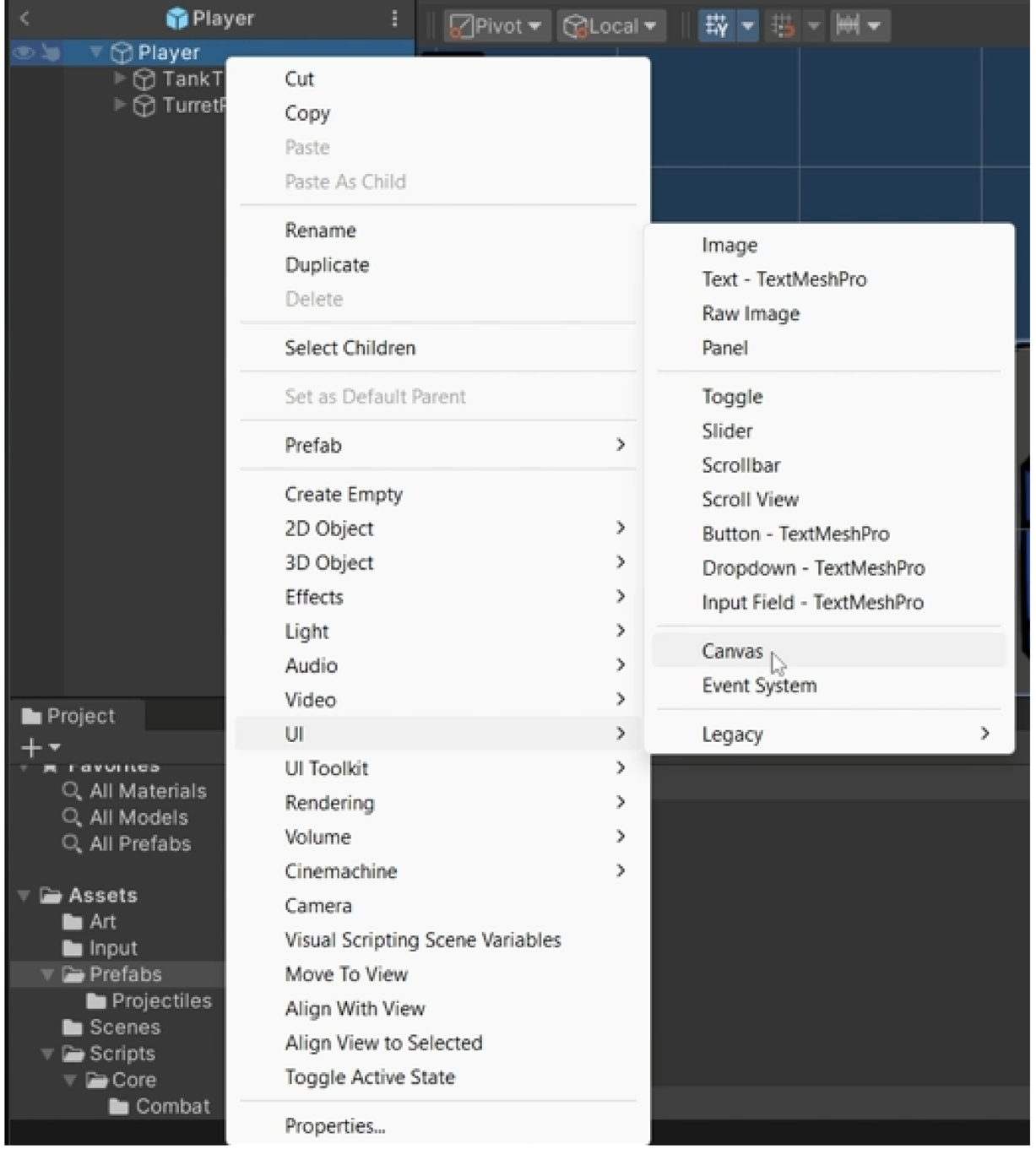
```
2 references
private void ModifyHealth(int value)
{
    if (isDead) { return; }

    int newHealth = CurrentHealth.Value + value;
    CurrentHealth.Value = Mathf.Clamp(newHealth, 0, MaxHealth);

    if(CurrentHealth.Value == 0)
    {
        OnDie?.Invoke(this);
        isDead = true;
    }
}
```

Health Component

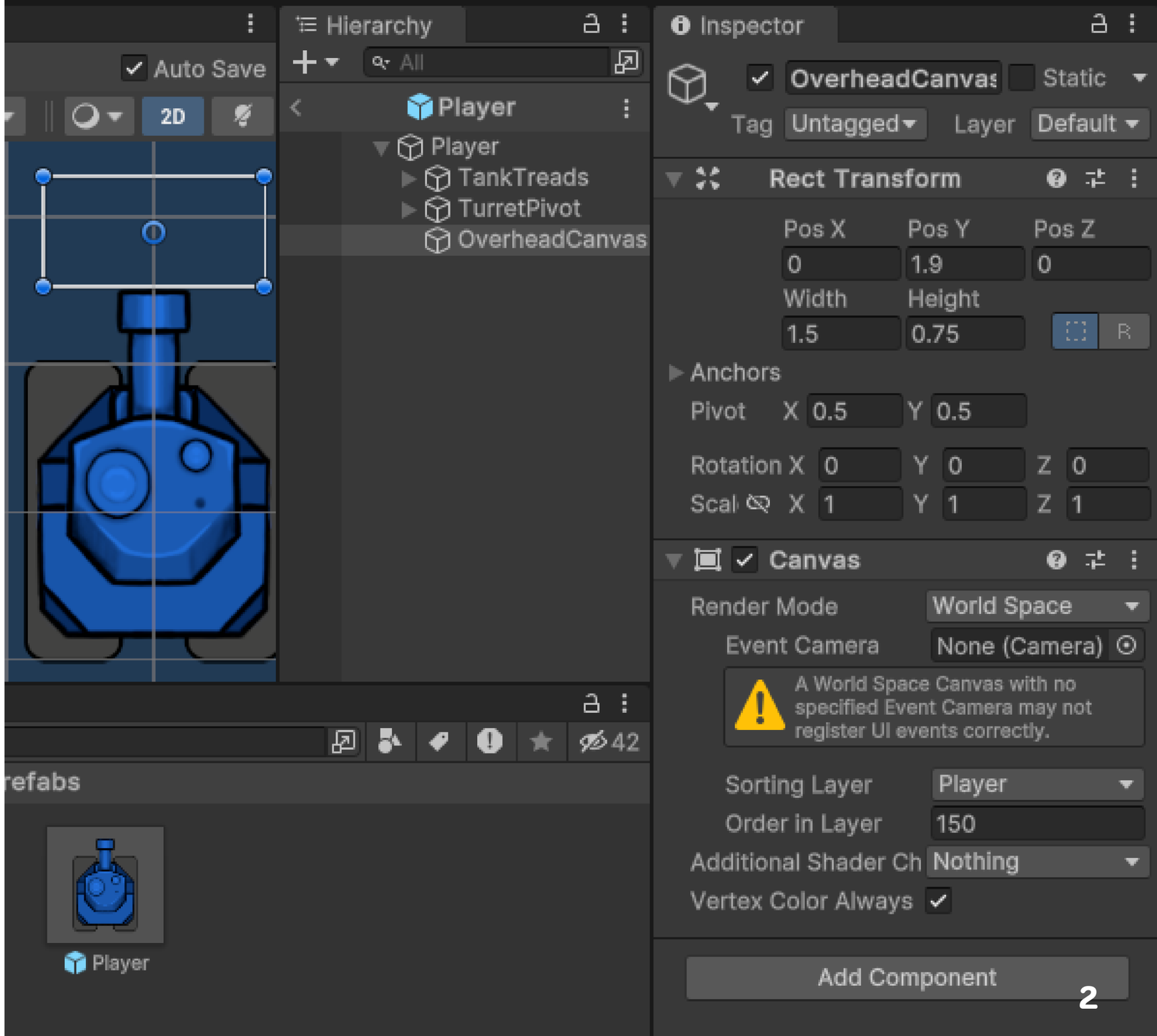


Health Display

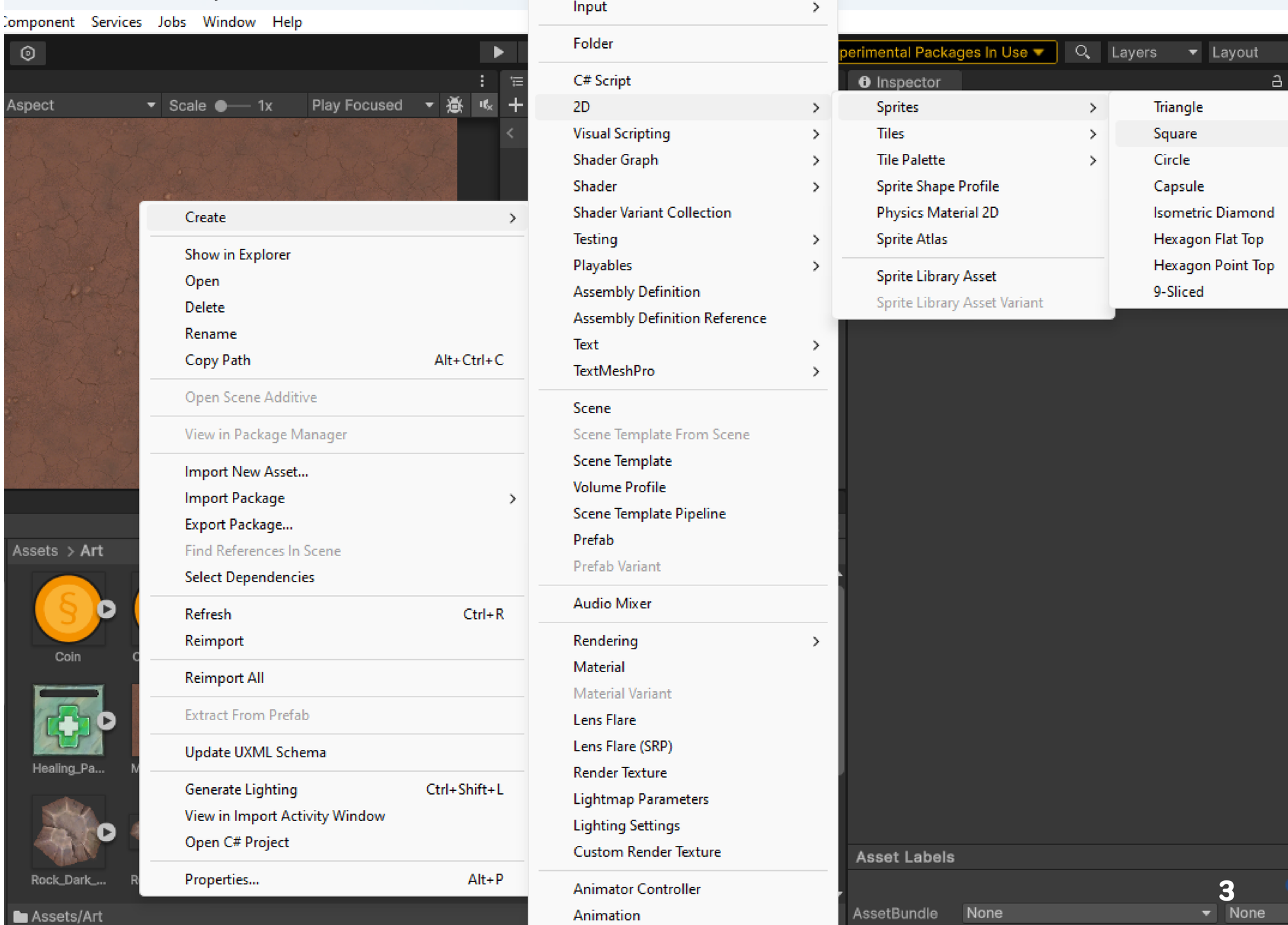
สร้าง Canvas ภายใน Prefab **Player**



เปลี่ยนชื่อ Canvas เป็น **OverheadCanvas**



สร้าง Square ไว้ใน โฟลเดอร์ Art

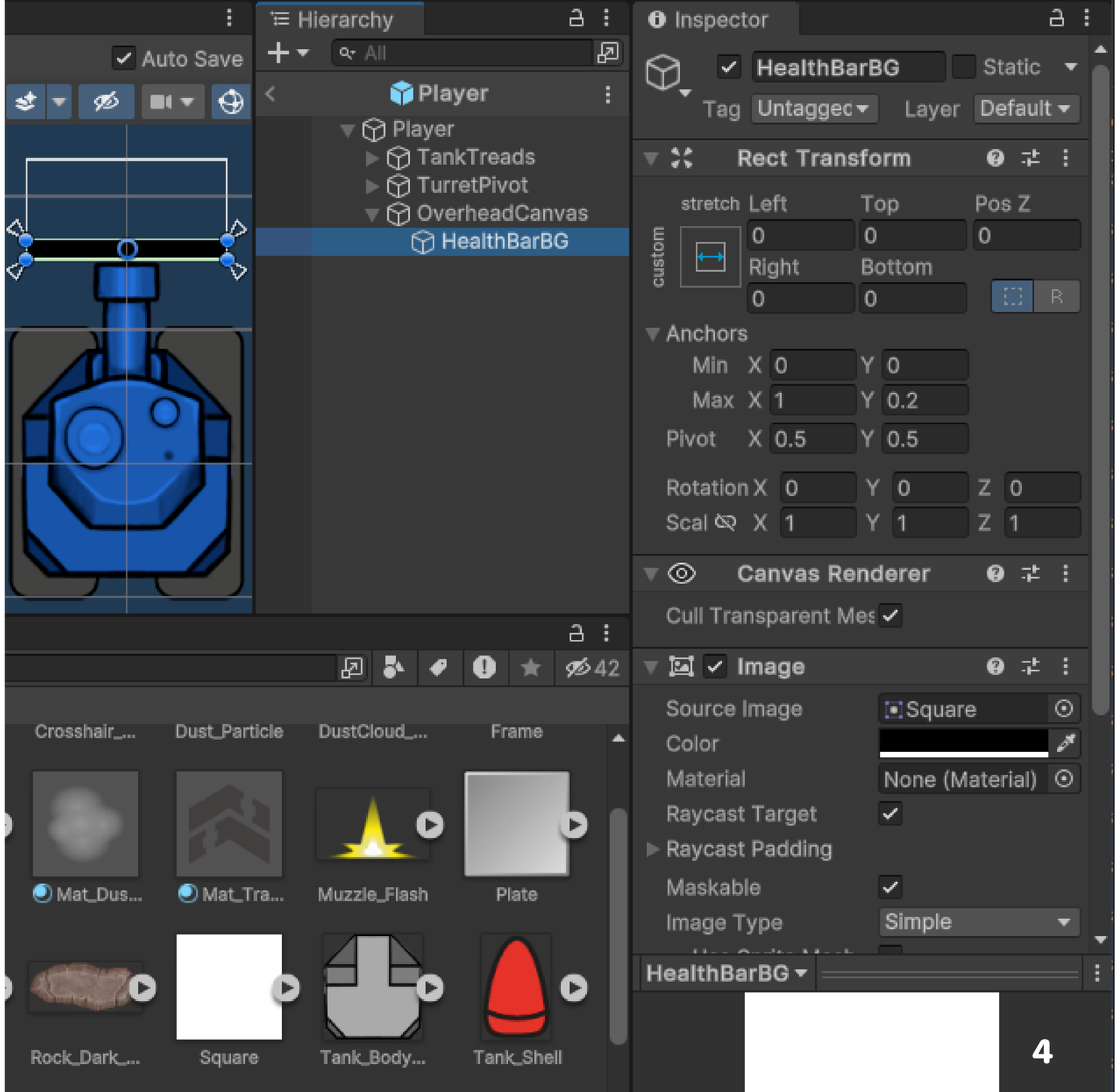


Health Display

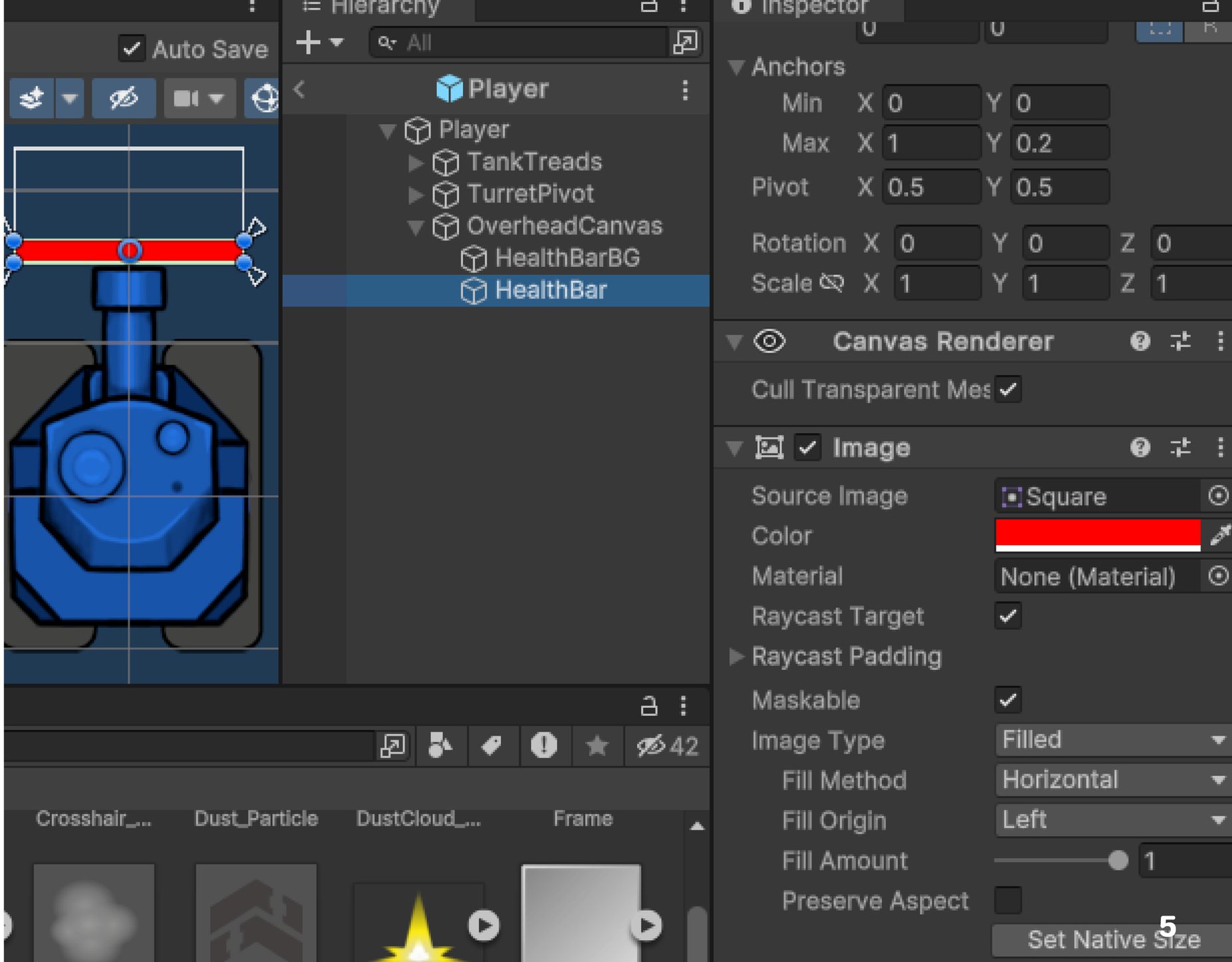
สร้าง Image แล้วตั้งชื่อว่า

HealthBarBG

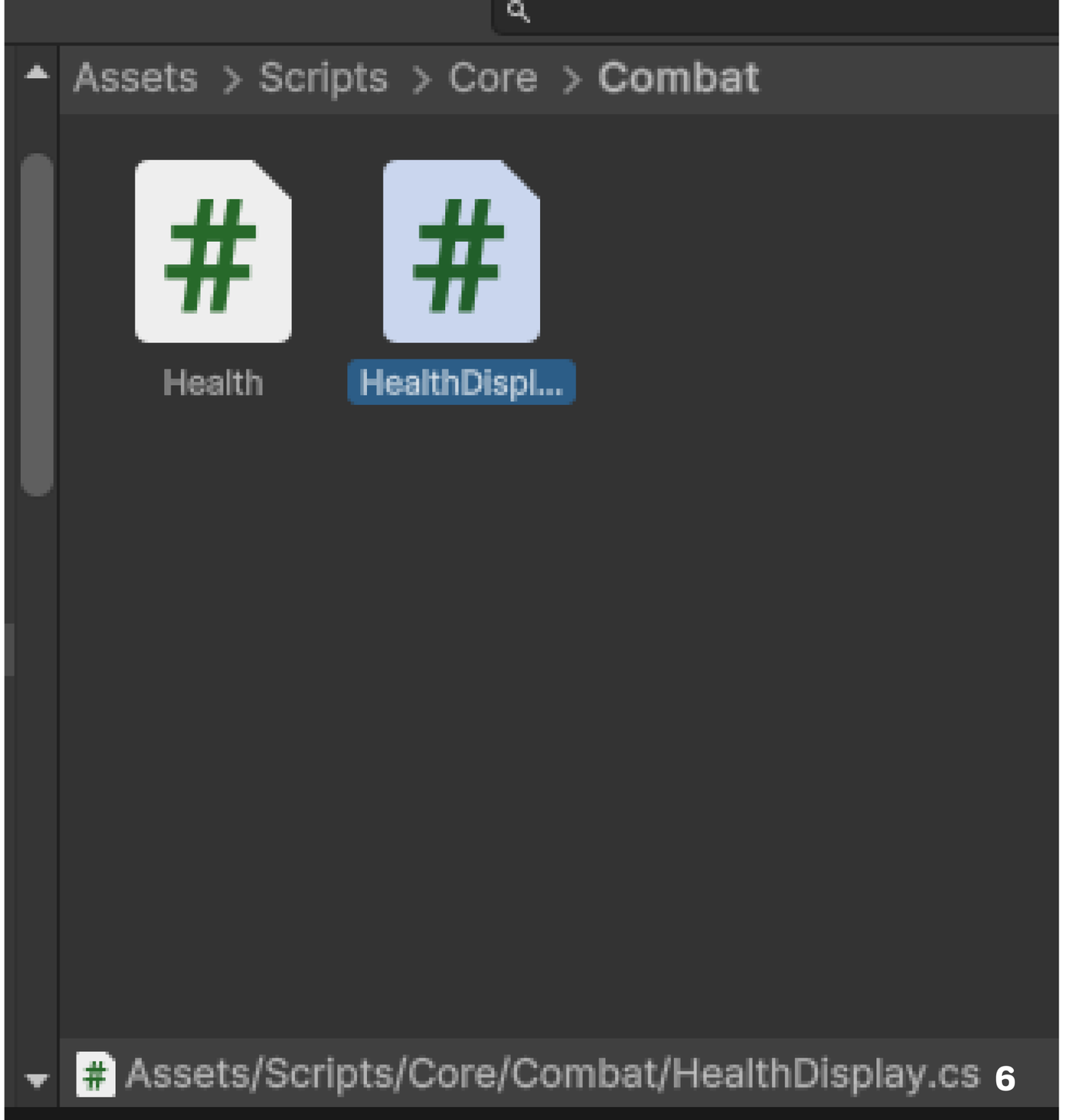
จากนั้นตั้งค่าตามในรูป



ทำเช่นเดียวกับอันก่อนหน้า
สร้าง Image แล้วตั้งชื่อว่า
HealthBar
จากนั้นตั้งค่าตามในรูป



เพิ่ม Script ชื่อว่า HealthDisplay



เพิ่มตัวแปรตามในภาพ

HealthDisplay.cs* | Health.cs

Assembly-CSharp | HealthDisplay

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
    0 references
6   public class HealthDisplay : NetworkBehaviour
7   {
8
9   }
10
```

```
[Header("References")]
0 references
[SerializeField] private Health health;
0 references
[SerializeField] private Image healthBarImage;
```

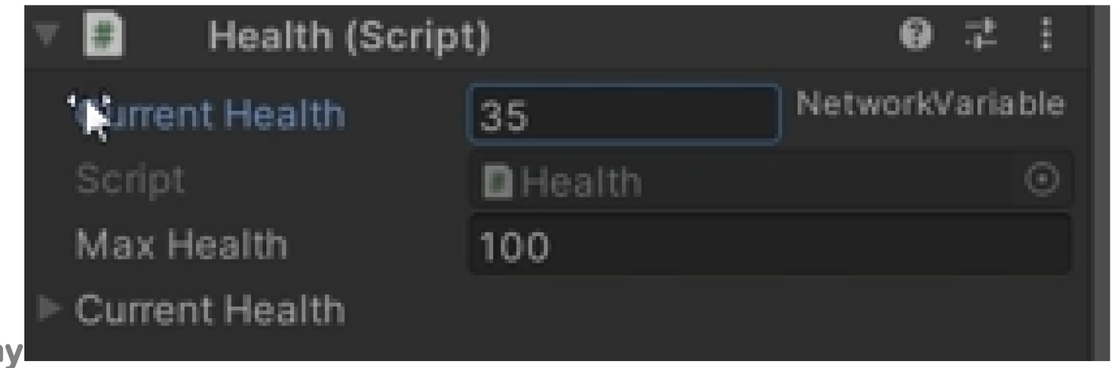
(field) Health HealthDisplay.health

สร้าง Functions

```
        1 reference
13      public override void OnNetworkSpawn()
14      {
15
16      }
17
        0 references
18      public override void OnNetworkDespawn()
19      {
20
21      }
22
        0 references
23      private void HandleHealthChanged(int oldHealth, int newHealth)
24      {
25
26      }
```

Modify Changed Event

- Create the **HandleHealthChanged(int, int)** method
- Subscribe and unsubscribe the method to/from **health.CurrentHealth.OnValueChanged**
- Only bother subscribing for clients
- Update the **healthBarImage.fillAmount** using the **newHealth** and **MaxHealth** values



Subscribe ให้กับ Event OnValueChanged

```
1 reference
public override void OnNetworkSpawn()
{
    if (!IsClient) { return; }

    health.CurrentHealth.OnValueChanged += HandleHealthChanged;
}
```

```
public override void OnNetworkDespawn()
{
    if (!IsClient) { return; }

    health.CurrentHealth.OnValueChanged -= HandleHealthChanged;
}
```

เมื่อค่าใน Network Variable มีการเปลี่ยนแปลง เราก็จะนำมาคำนวณและอีซัพไปที่ UI ของ HP

```
private void HandleHealthChanged(int oldHealth, int newHealth)
{
    healthBarImage.fillAmount = (float)newHealth / health.MaxHealth;
}
```

NetworkVariable | Unity Multiplayer Networking

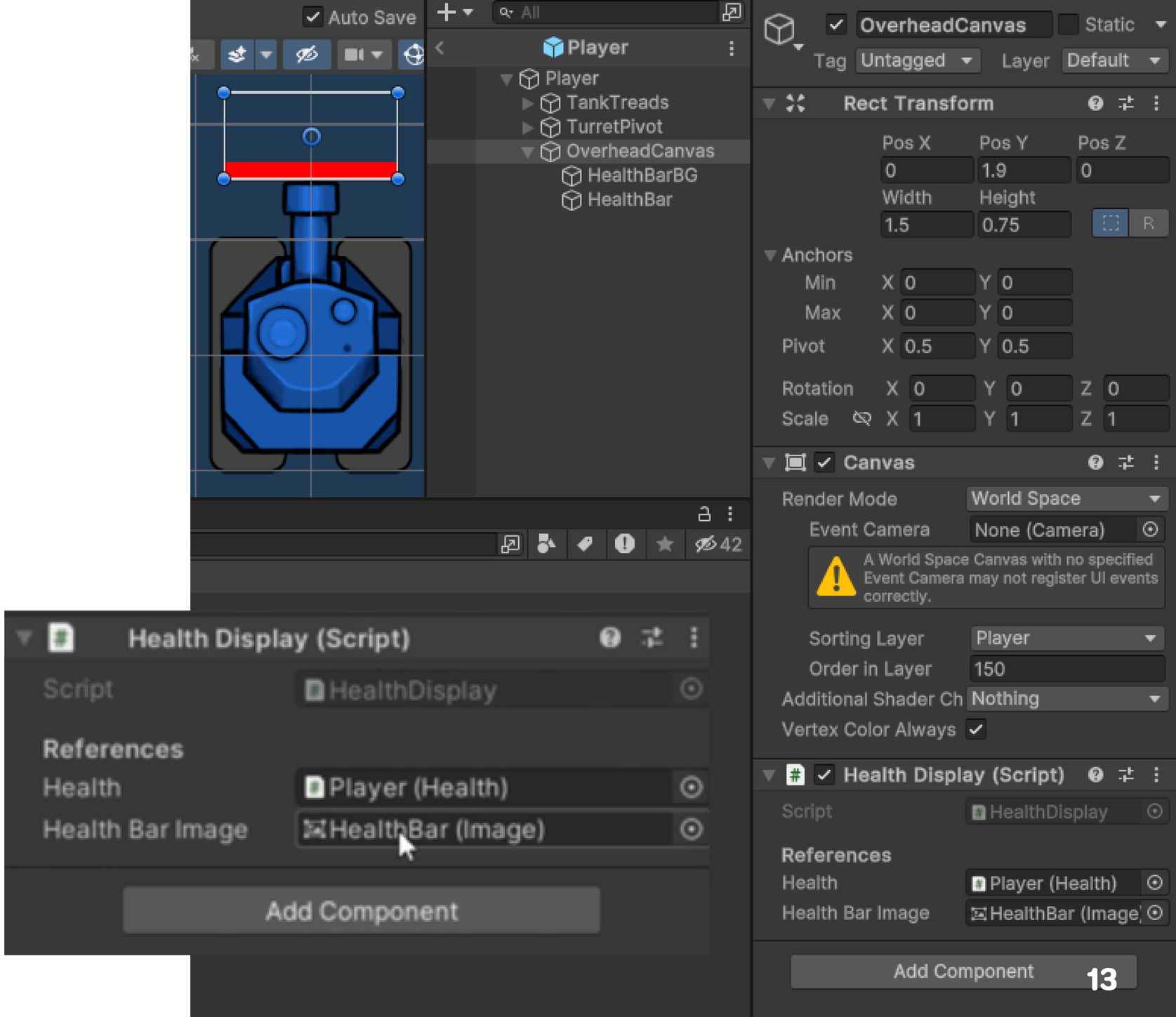
Introduction

unity3d.com

```
1 reference
public override void OnNetworkSpawn()
{
    if (!IsClient) { return; }

    health.CurrentHealth.OnValueChanged += HandleHealthChanged;
    HandleHealthChanged(0, health.CurrentHealth.Value);
}
```

ทำการใส่ค่าเริ่มต้นให้กับ CurrentHealth

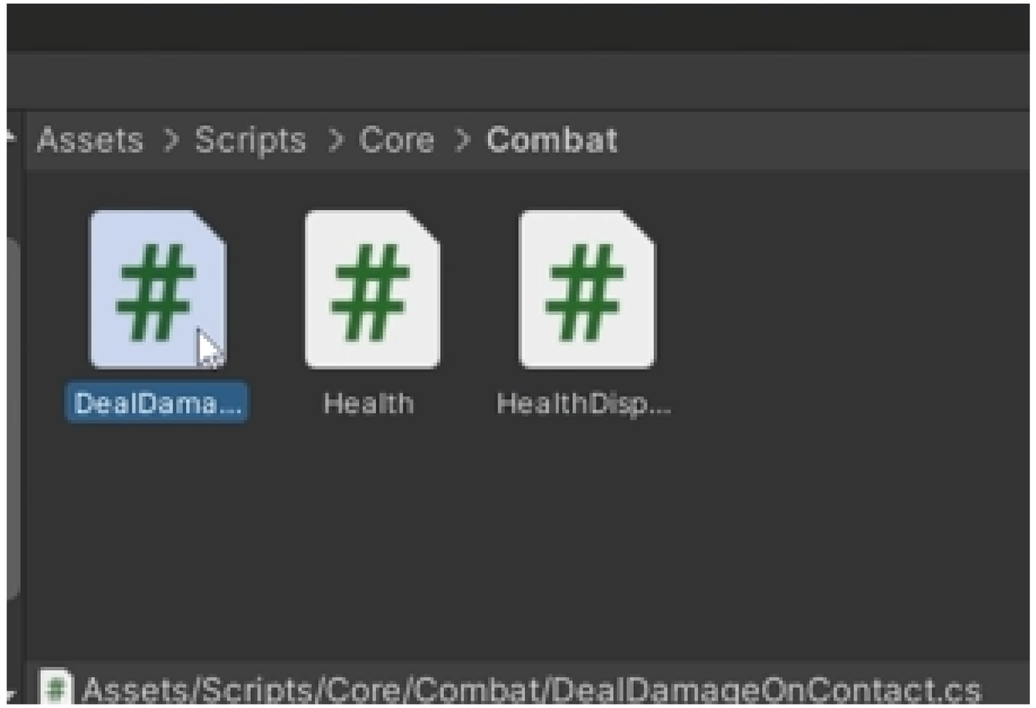
ตั้งค่าให้กับ Script HealthDisplay



Health Display

Dealing Damage

สร้าง Script DealDamageOnContact



DealDamageOnContact

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity Script | 0 references
public class DealDamageOnContact : MonoBehaviour
{
    [SerializeField] private int damage = 5;
}
```

Dealing Damage

- Create **OnTriggerEnter2D(Collider2D)** method
- Make sure the collider has an attachedRigidbody
- If it does, then try getting the **Health** component from it
- If you successfully get the **Health** component, call our **TakeDamage(int)** method

เรียกใช้ function OnTriggerEnter2D

```
0 references
private void OnTriggerEnter2D(Collider2D col)
{
    if (col.attachedRigidbody == null) { return; }

    if(col.attachedRigidbody.TryGetComponent<Health>(out Health health))
    {
        health.TakeDamage(damage);
    }
}
```

(parameter) Collider2D col

สร้าง Function **SetOwner**

```
0 references
public class DealDamageOnContact : MonoBehaviour
{
    1 reference
    [SerializeField] private int damage = 5;

    1 reference
    private ulong ownerClientId;

    0 references
    public void SetOwner(ulong ownerClientId)
    {
        this.ownerClientId = ownerClientId;
    }
```

DealDamageOnContact.cs* | HealthDisplay.cs | Health.cs

Assembly-CSharp | DealDamag

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
```

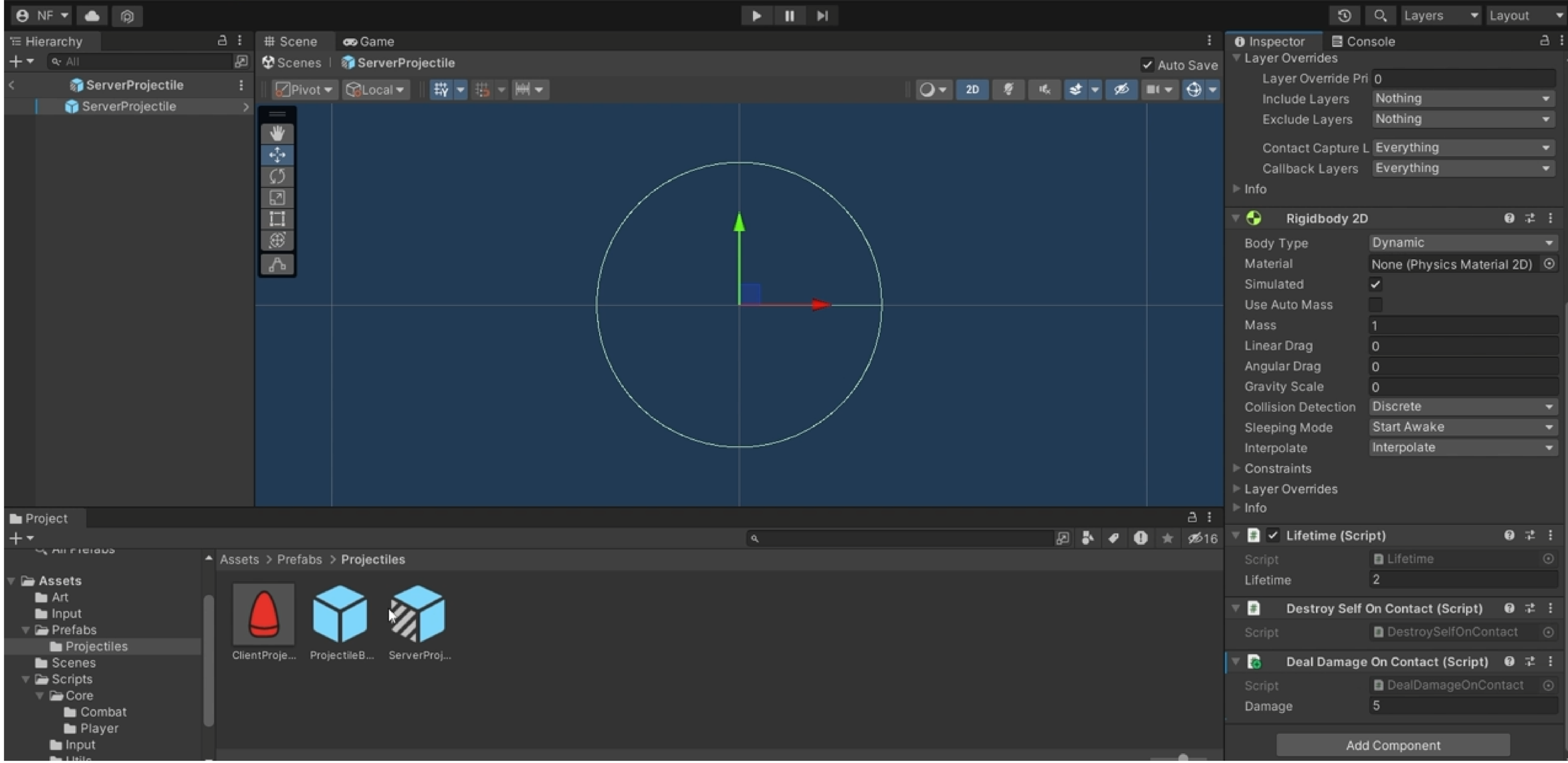
เช็คว่าลูกกระสุนที่ชนไม่ใช่จะเจ้าของเดียวกัน ถ้าตรงกันให้ทะลุผ่าน ถ้าต่างกันหรืออื่นๆ ให้สร้าง Damage ได้ถ้า Obj มี component Health

```
Unity Message | 0 references
private void OnTriggerEnter2D(Collider2D col)
{
    if(col.attachedRigidbody == null) {  return; }

    if(col.attachedRigidbody.TryGetComponent<NetworkObject>(out NetworkObject netObj))
    {
        if(ownerClientId == netObj.OwnerClientId)
        {
            return;
        }
    }

    if(col.attachedRigidbody.TryGetComponent<Health>(out Health health))
    {
        health.TakeDamage(damage);
    }
}
```

ใน Script ProjectileLauncher

```
Unity Script (1 asset reference) | 0 references
public class ProjectileLauncher : NetworkBehaviour
{
```

```
1 reference
private void PrimaryFireServerRpc(Vector3 spawnPos, Vector3 direction)
{
    GameObject projectileInstance = Instantiate(
        serverProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;

    Physics2D.IgnoreCollision(playerCollider, projectileInstance.GetComponent<Collider2D>());

    if(projectileInstance.TryGetComponent<DealDamageOnContact>(out DealDamageOnContact dealDamage))
    {
        dealDamage.SetOwner(OwnerClientId);
    }

    if (projectileInstance.TryGetComponent<Rigidbody2D>(out Rigidbody2D rb))
    {
        rb.velocity = rb.transform.up * projectileSpeed;
    }

    SpawnDummyProjectileClientRpc(spawnPos, direction);
}
```



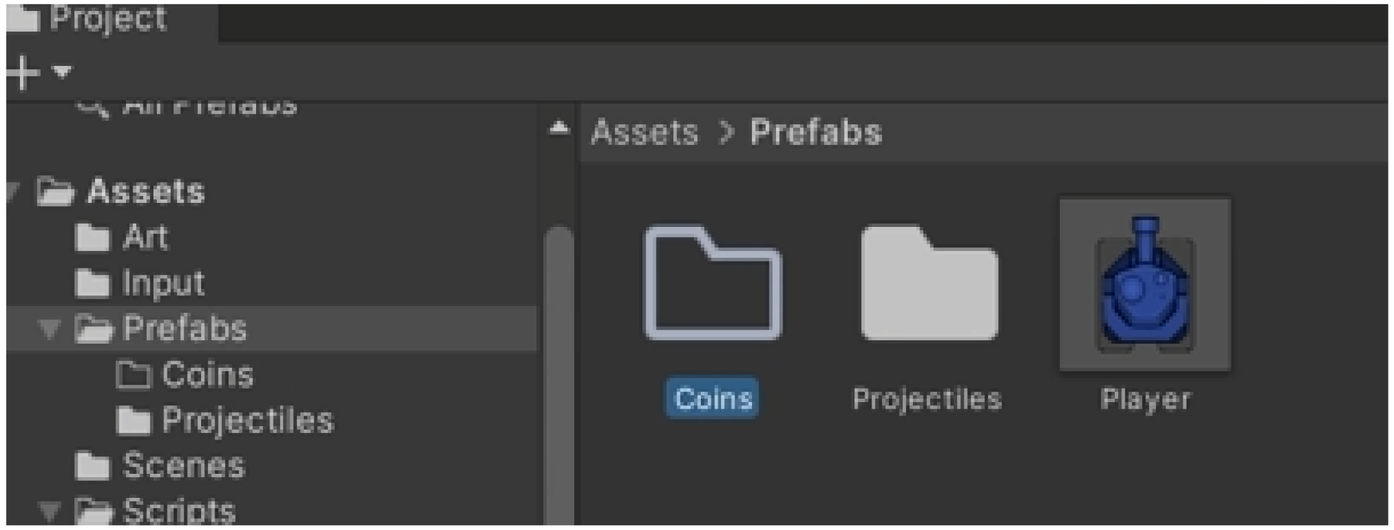
นำ component ที่ทำไว้ใส่ใน ServerProjectile

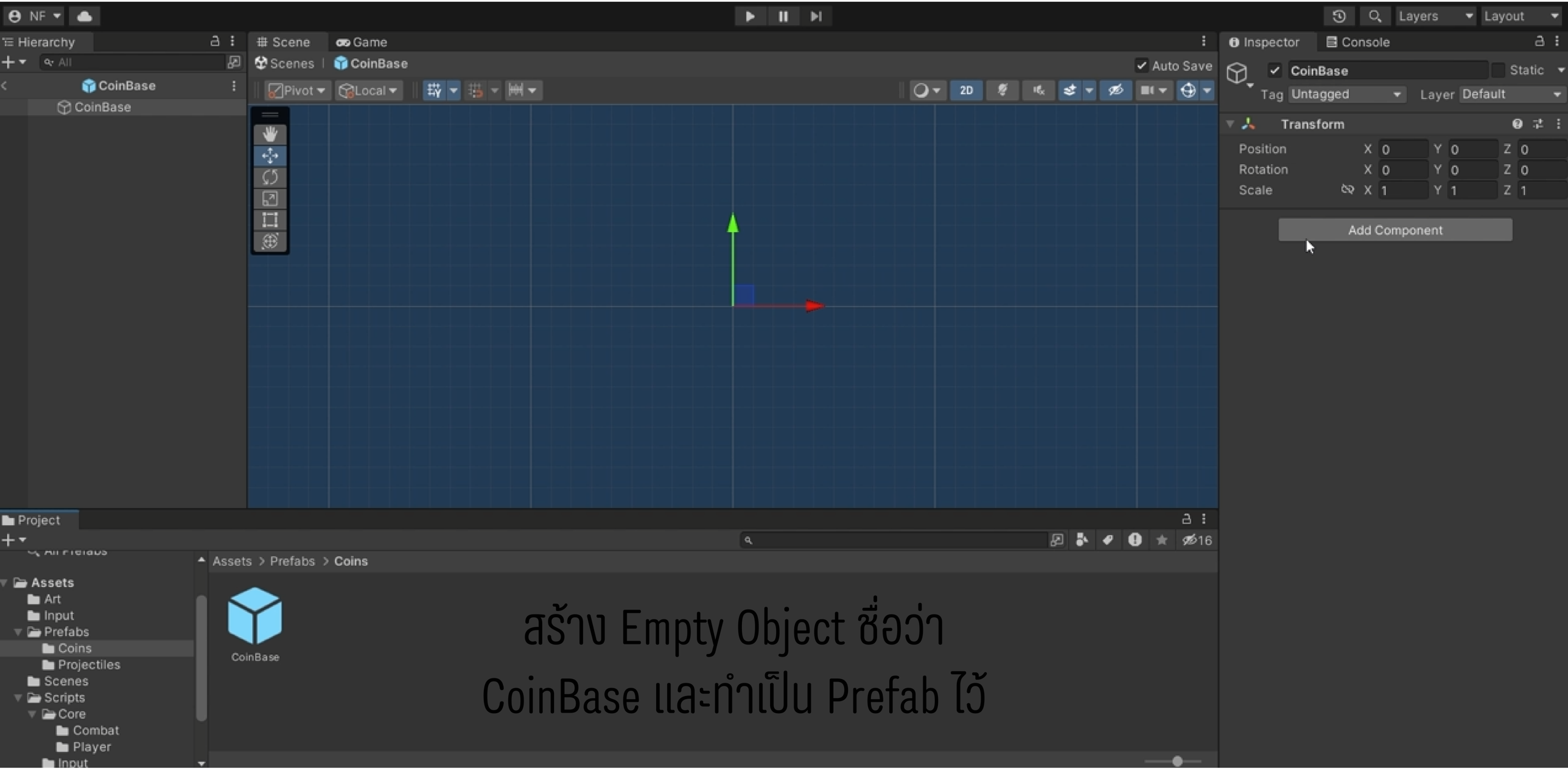
Dealing Damage

Coins

สร้าง Folder

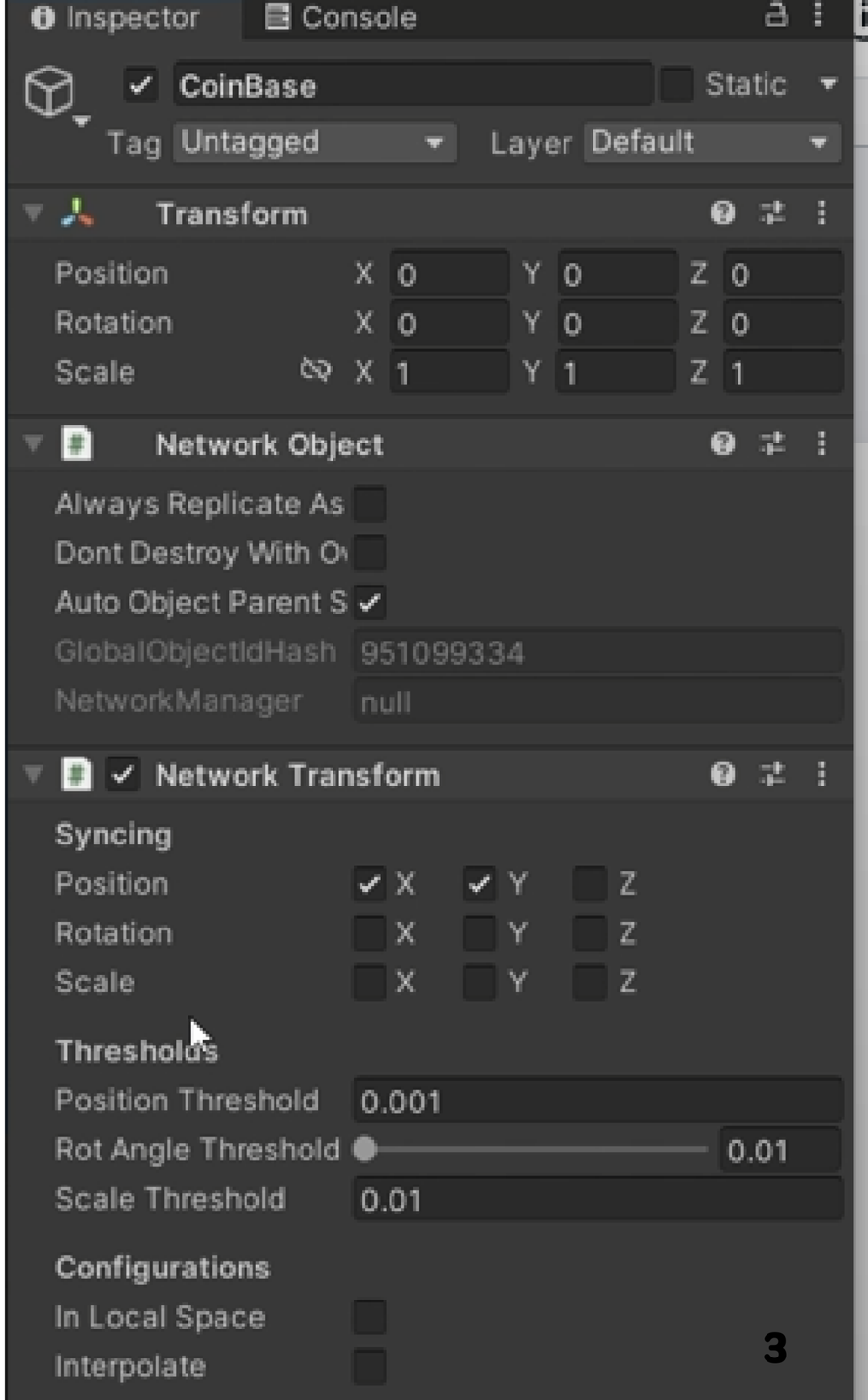
Coins



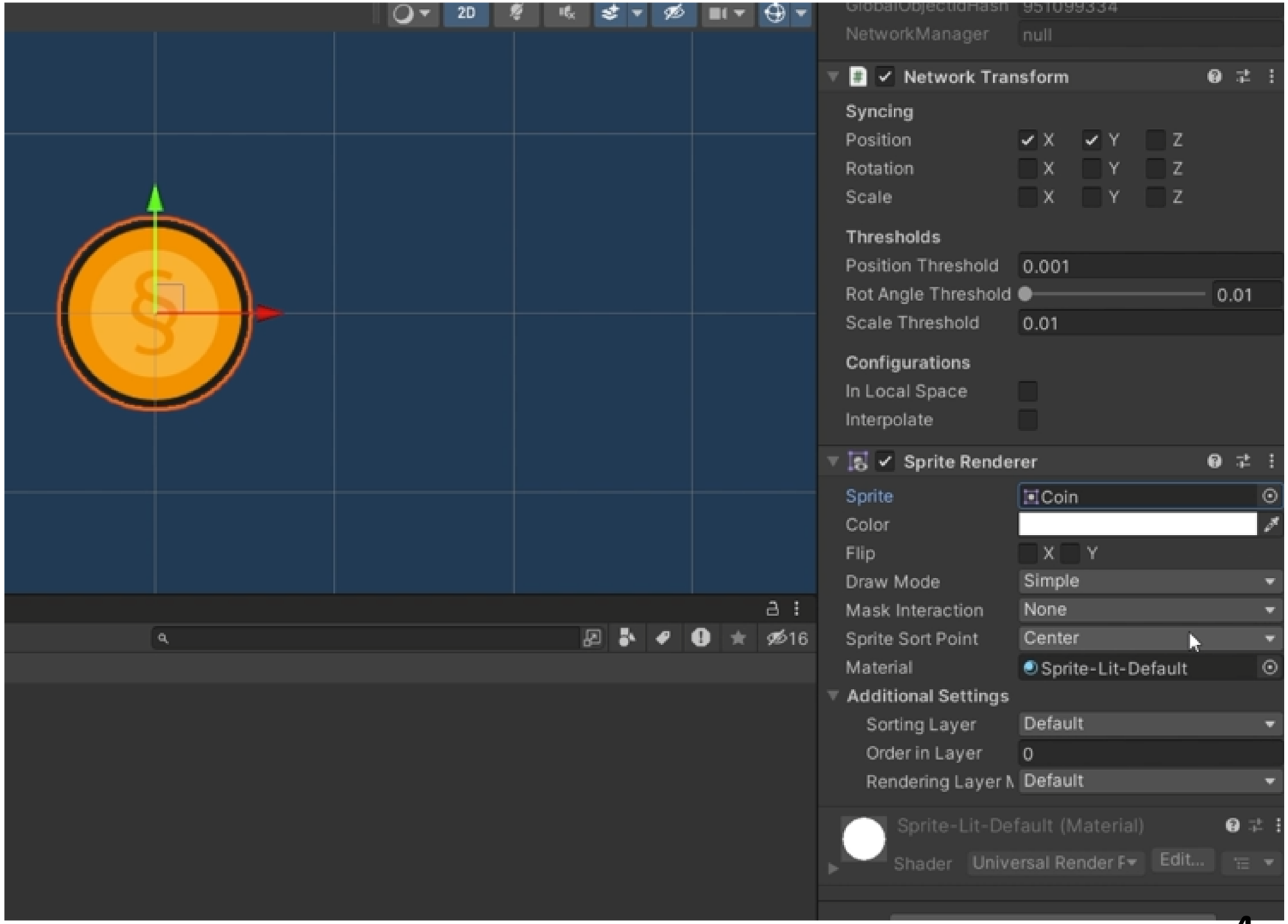


สร้าง Empty Object ชื่อว่า CoinBase และทำเป็น Prefab ไว้

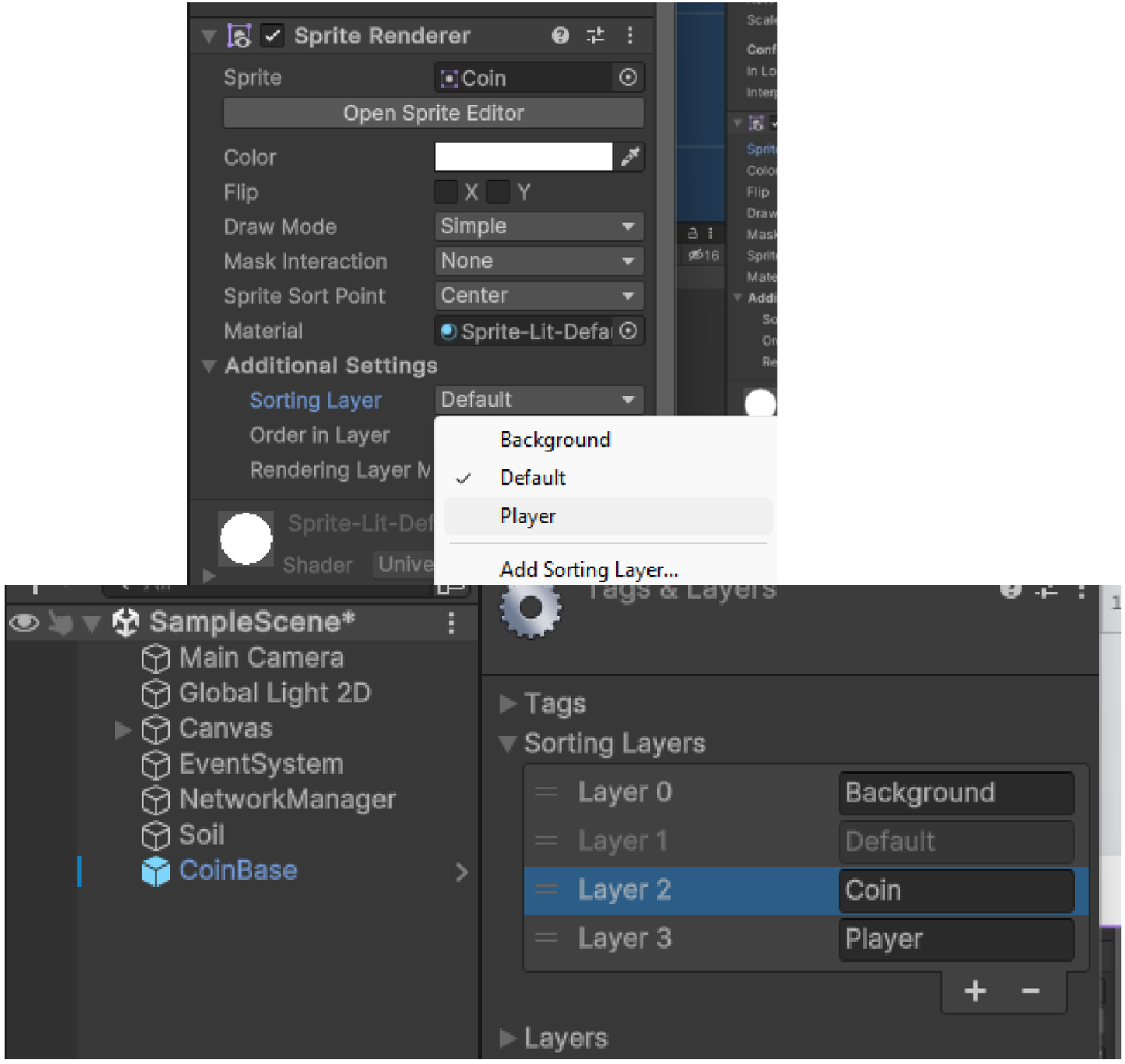
ใส่ Component
NetworkObject และ
NetworkTransform

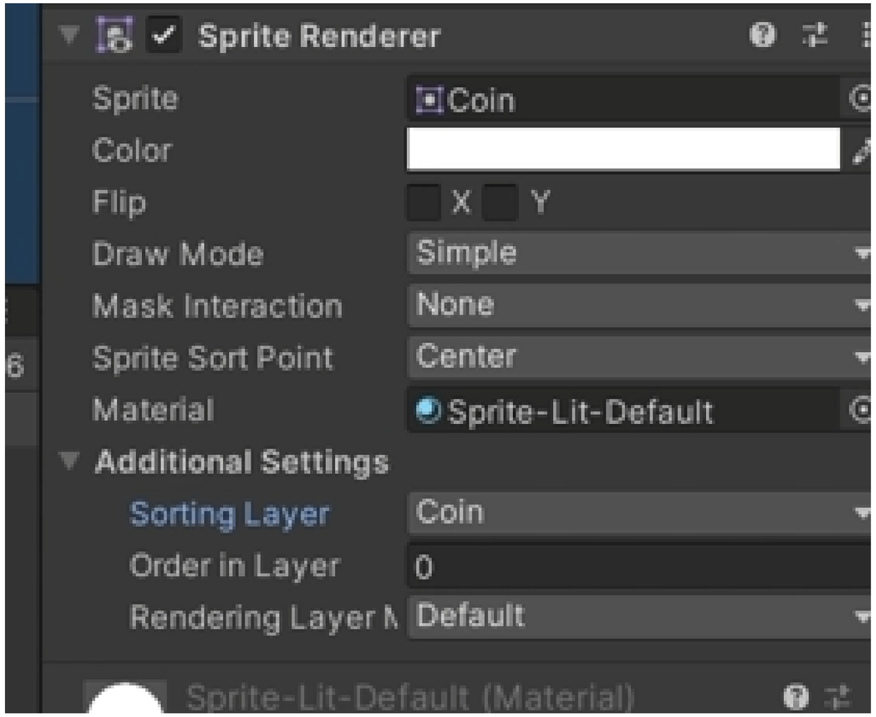


ใส่ SpriteRenderer

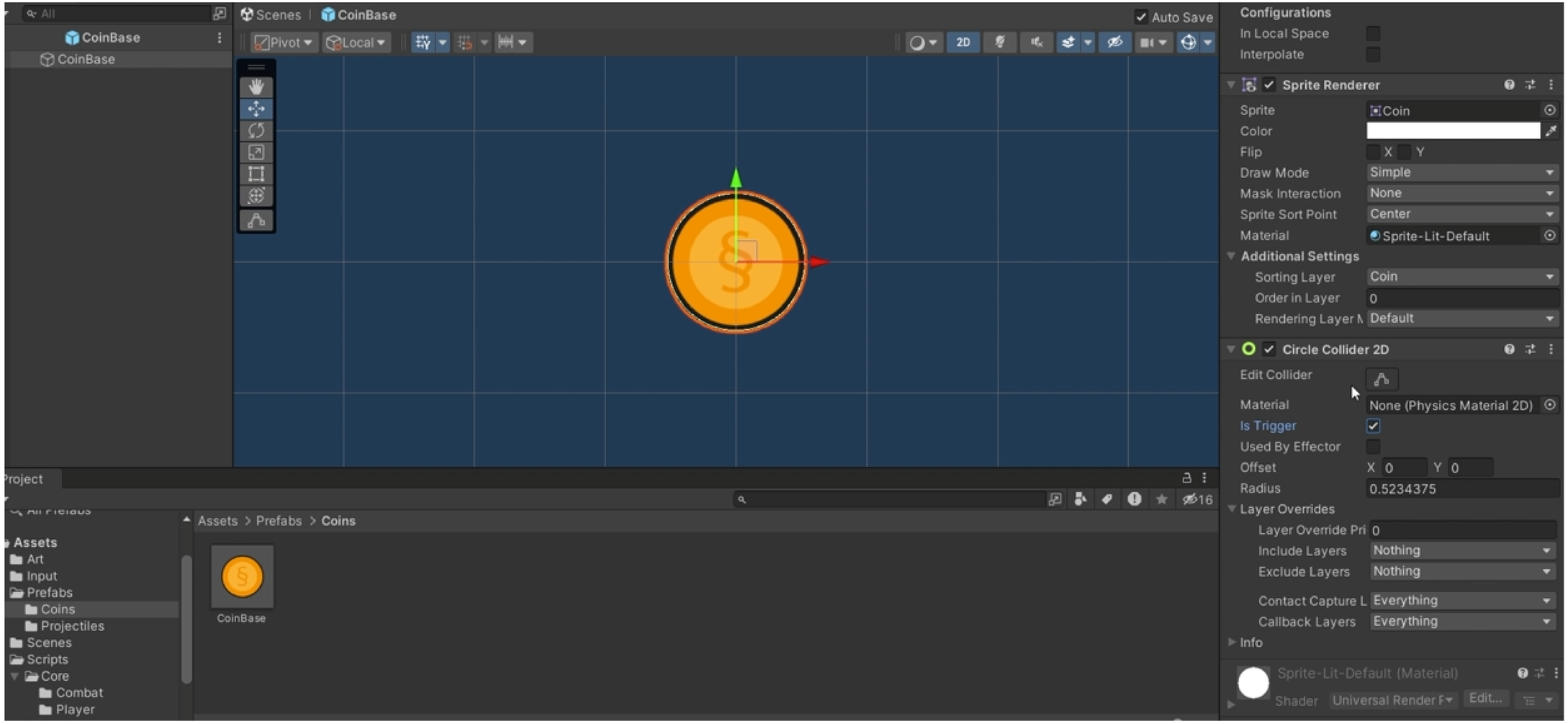


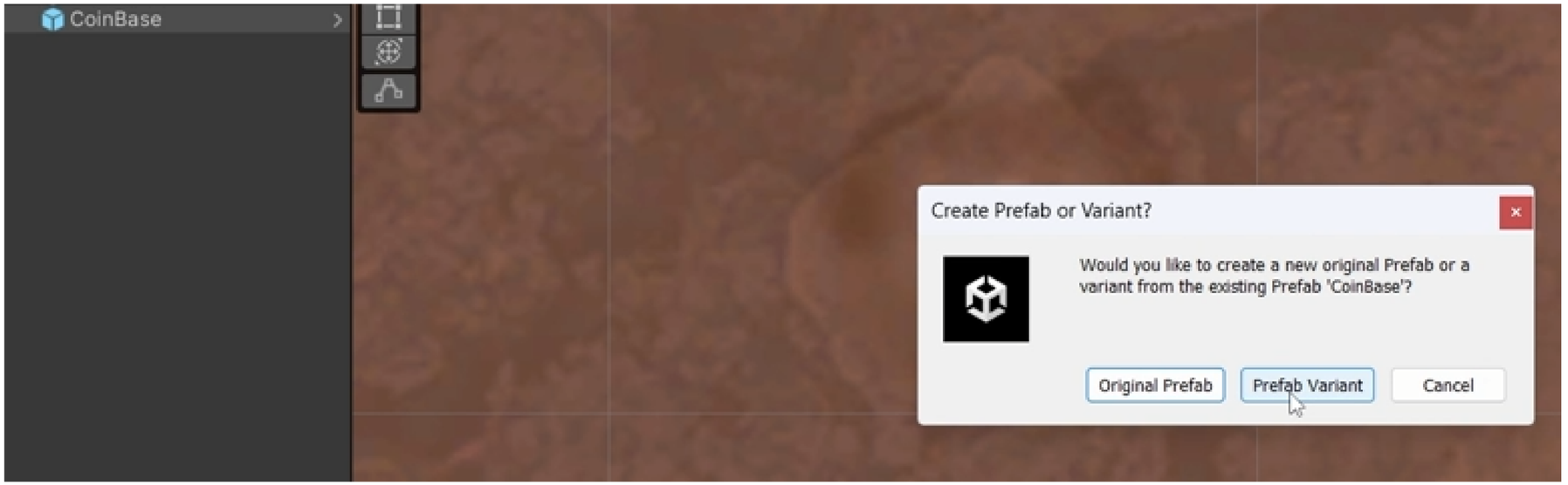
ตั้งค่า Layer Coin ให้อยู่
หลัง Player



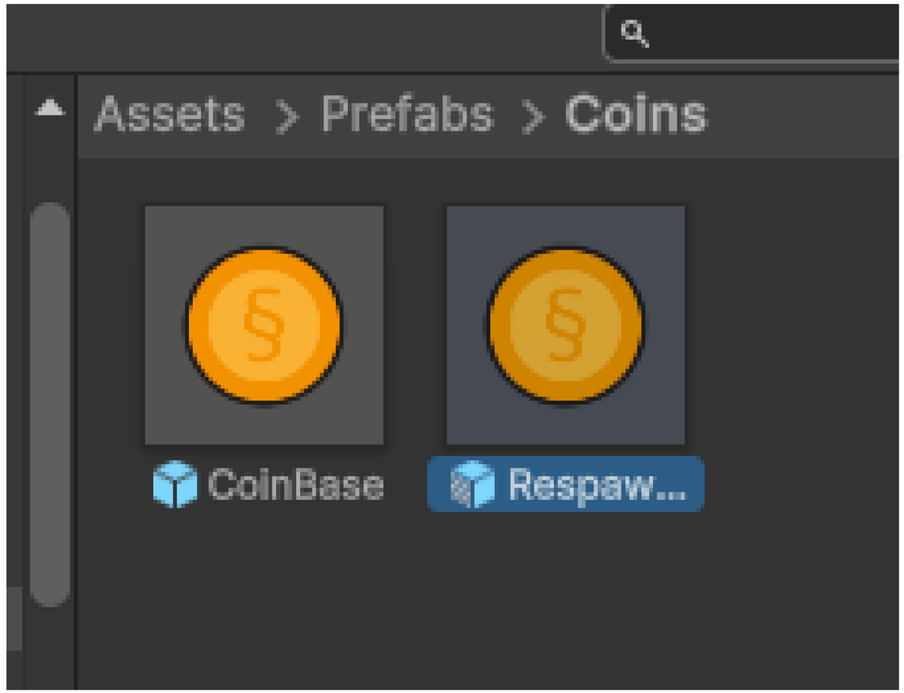


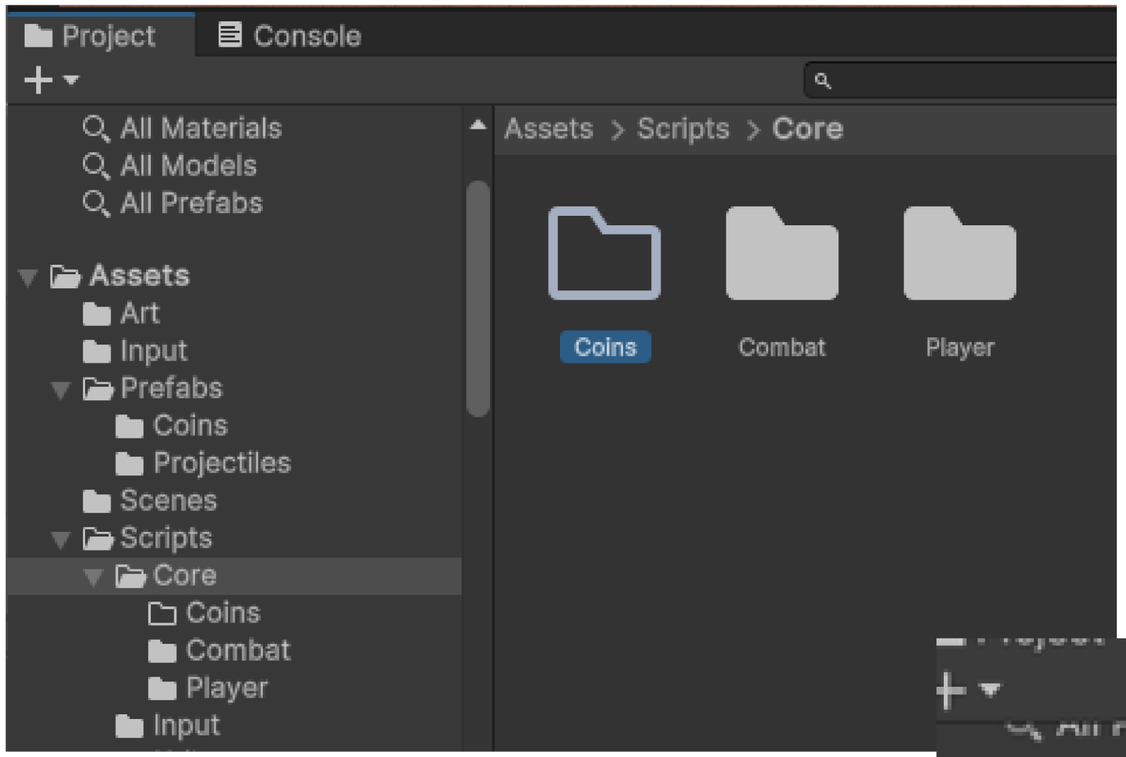
ใส่ CircleCollider2D

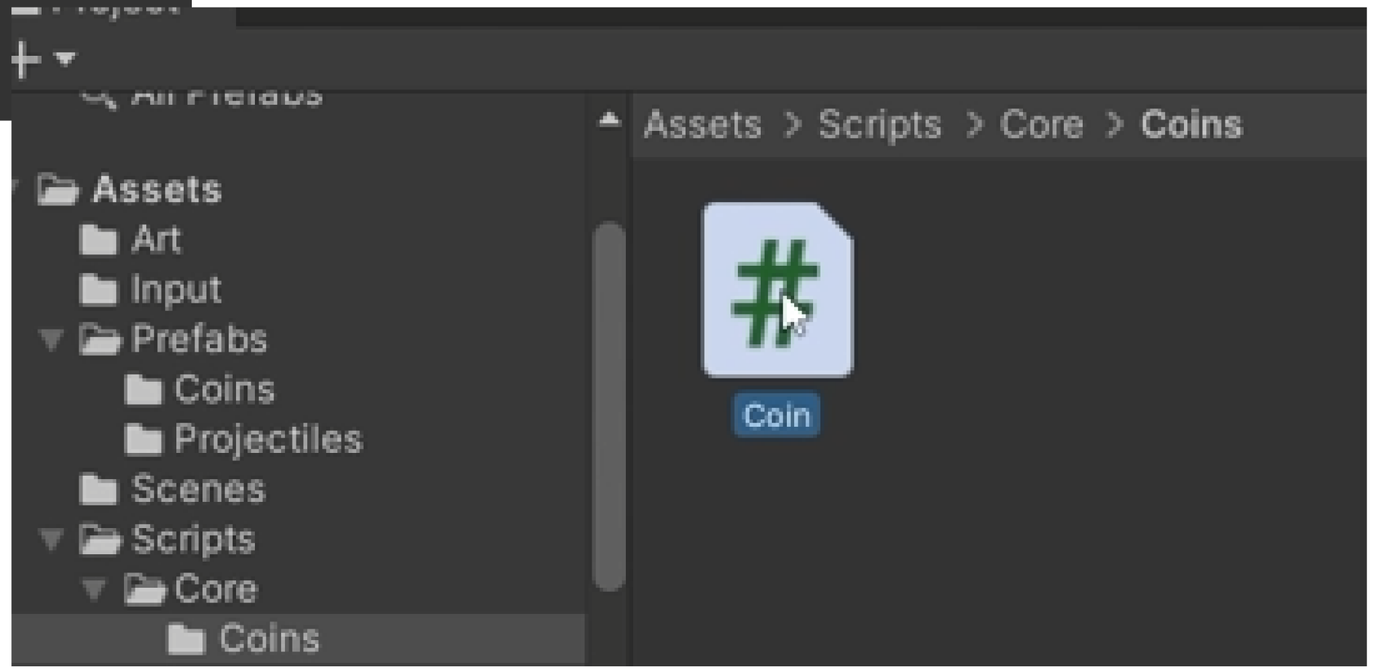




สร้างเป็น Prefab Variant
ชื่อว่า RespawningCoin

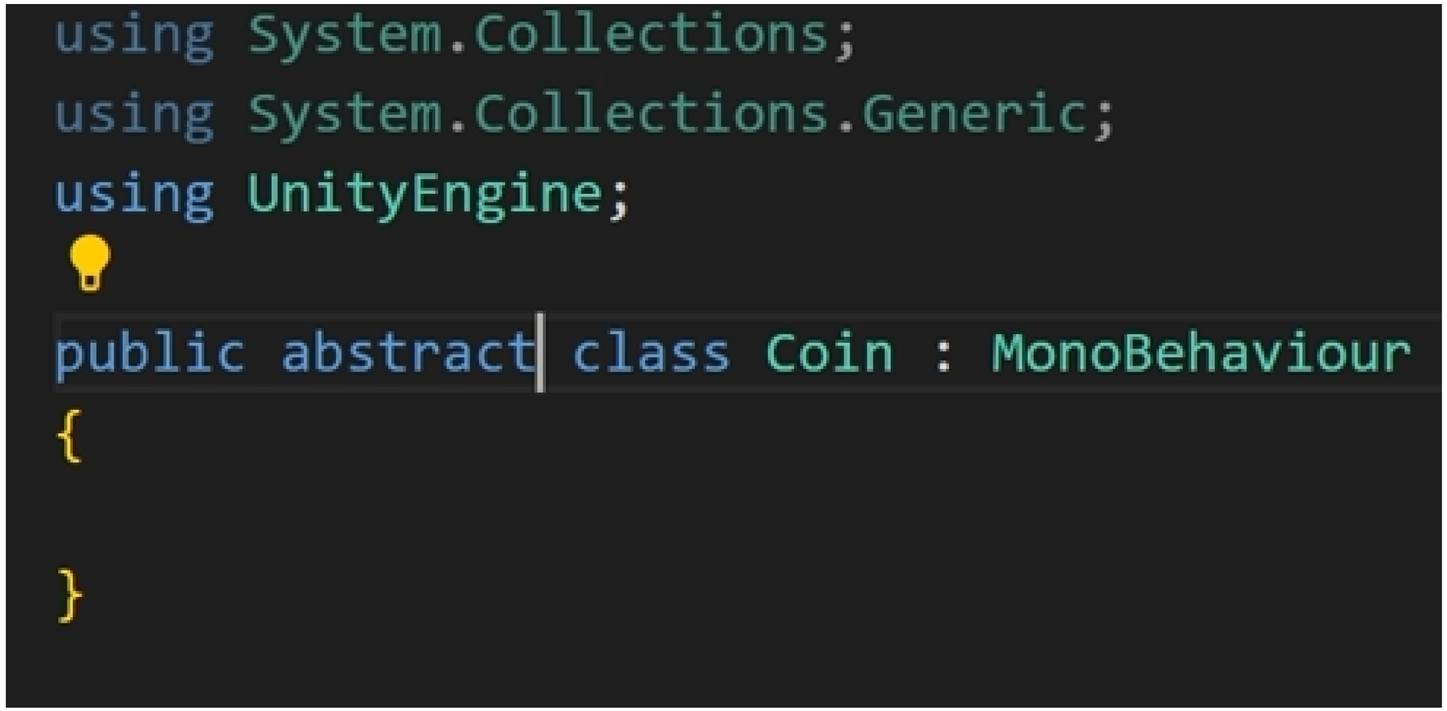




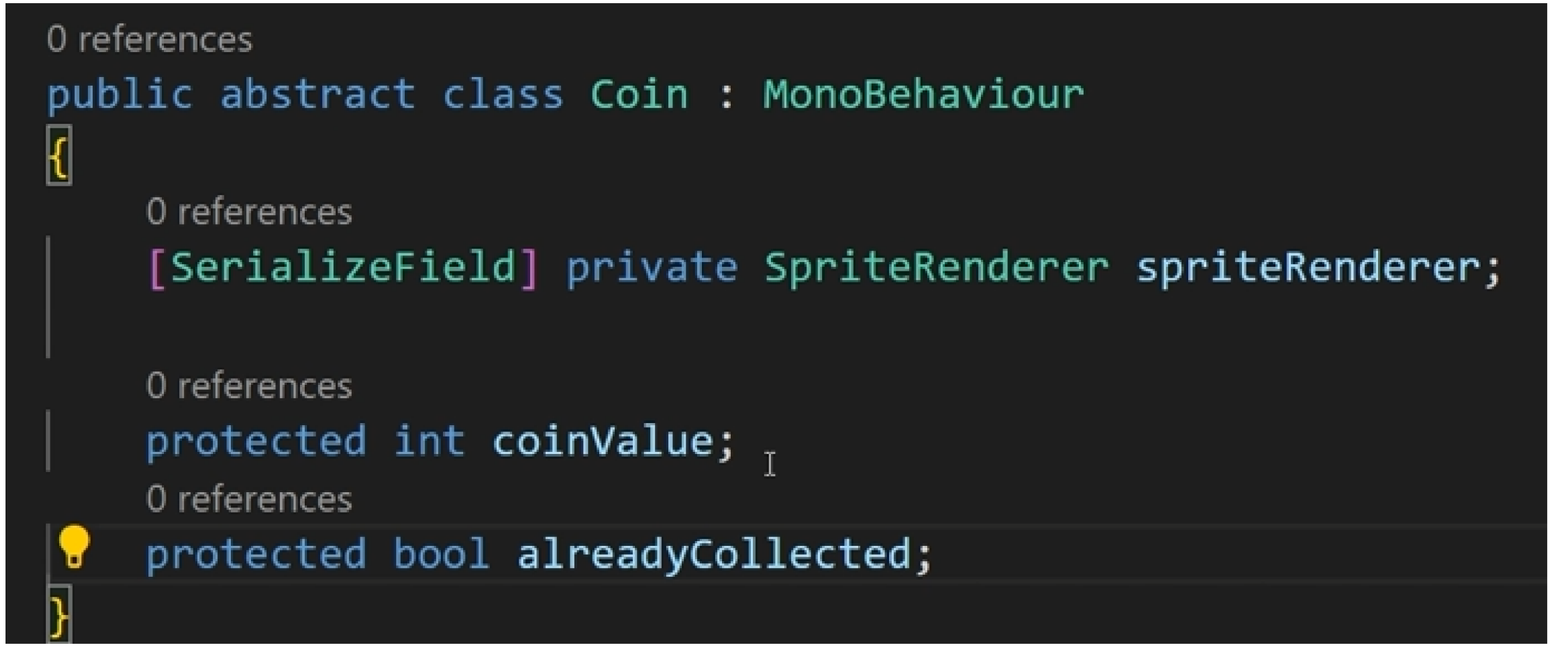


สร้าง Folder Coins

สร้าง Script Coin



Abstract Class = This class itself can never exist, Only children of this can exist.



Why use **protected?**

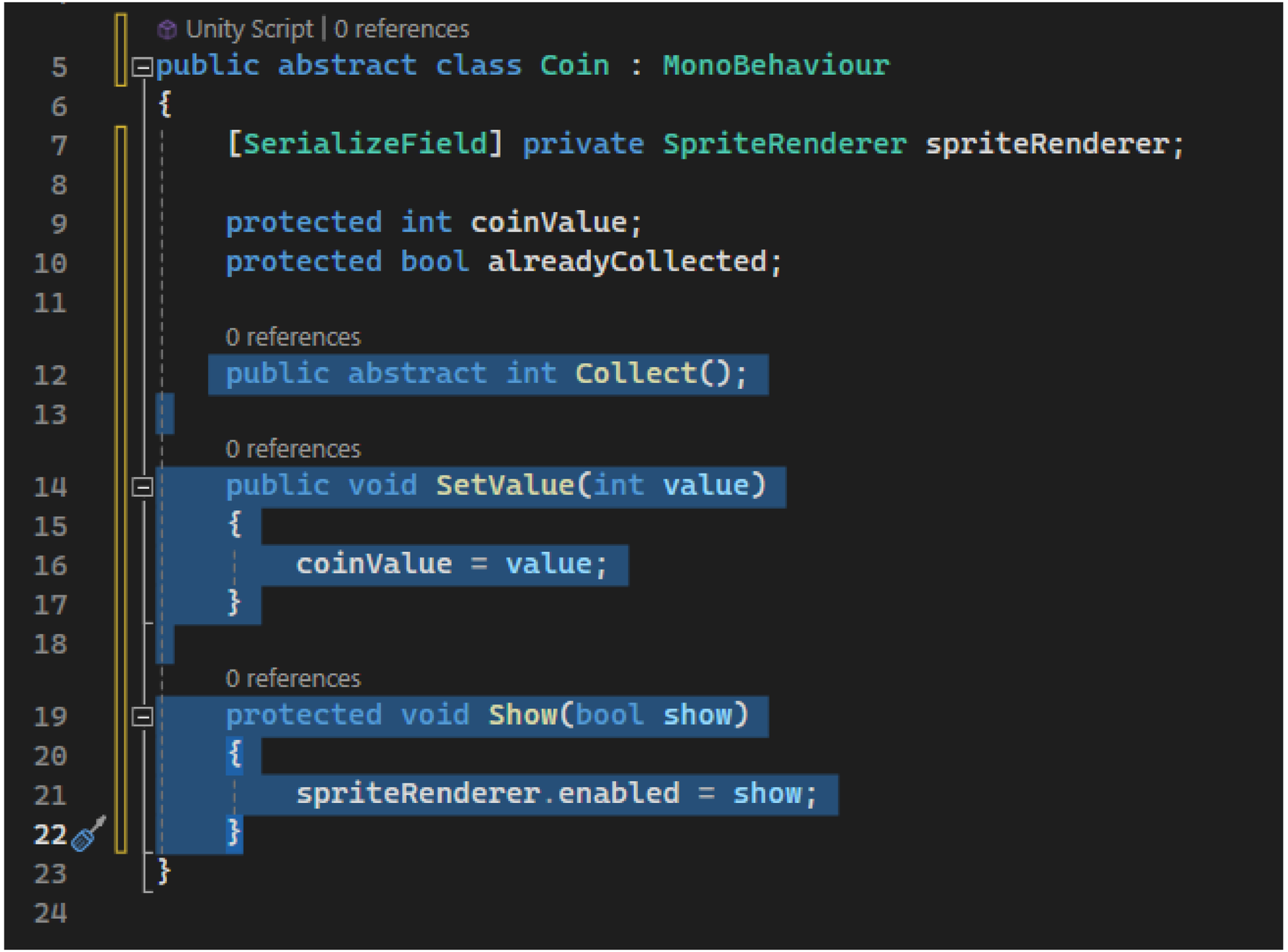
```
0 references
public abstract class Coin : MonoBehaviour
{
    0 references
    [SerializeField] private SpriteRenderer spriteRenderer;

    0 references
    protected int coinValue;
    0 references
    protected bool alreadyCollected;
}
```

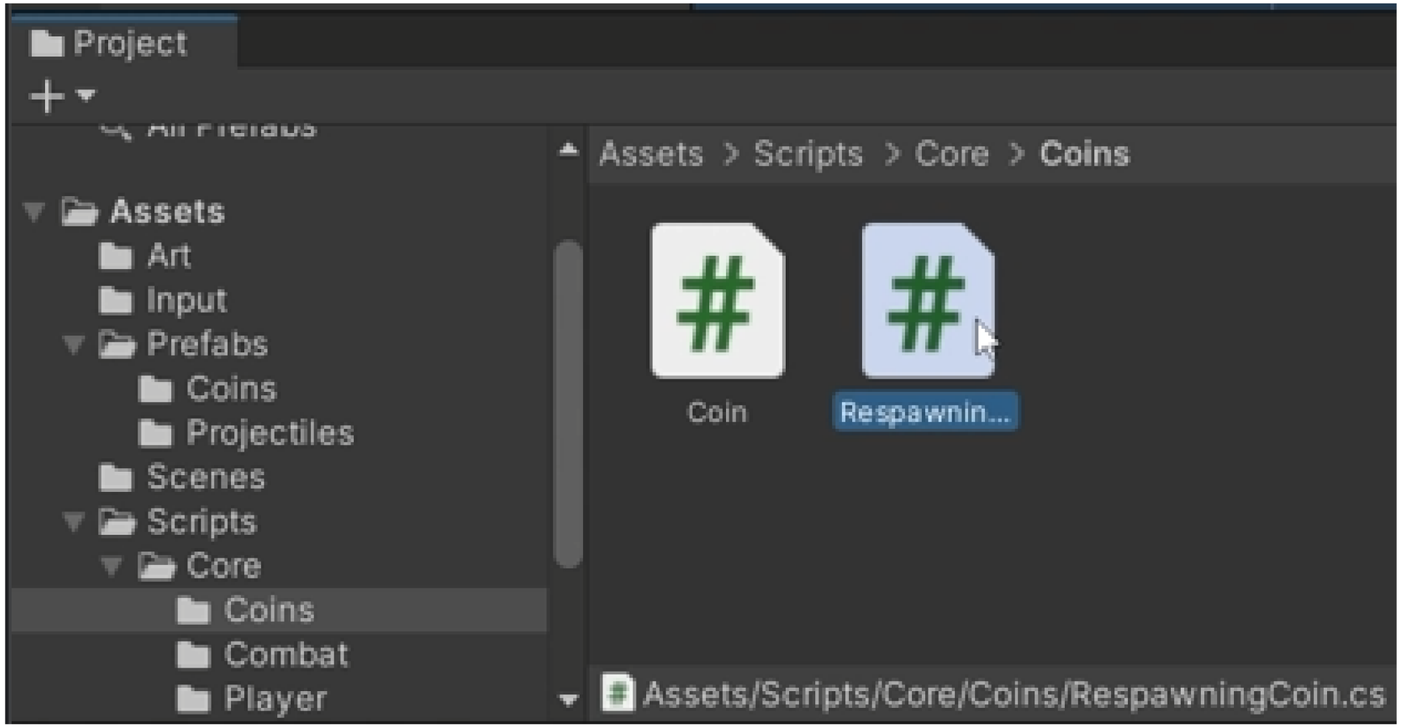
Why use **protected?**

Because there could be a scenario where two player run over the coin at the exact same frame.

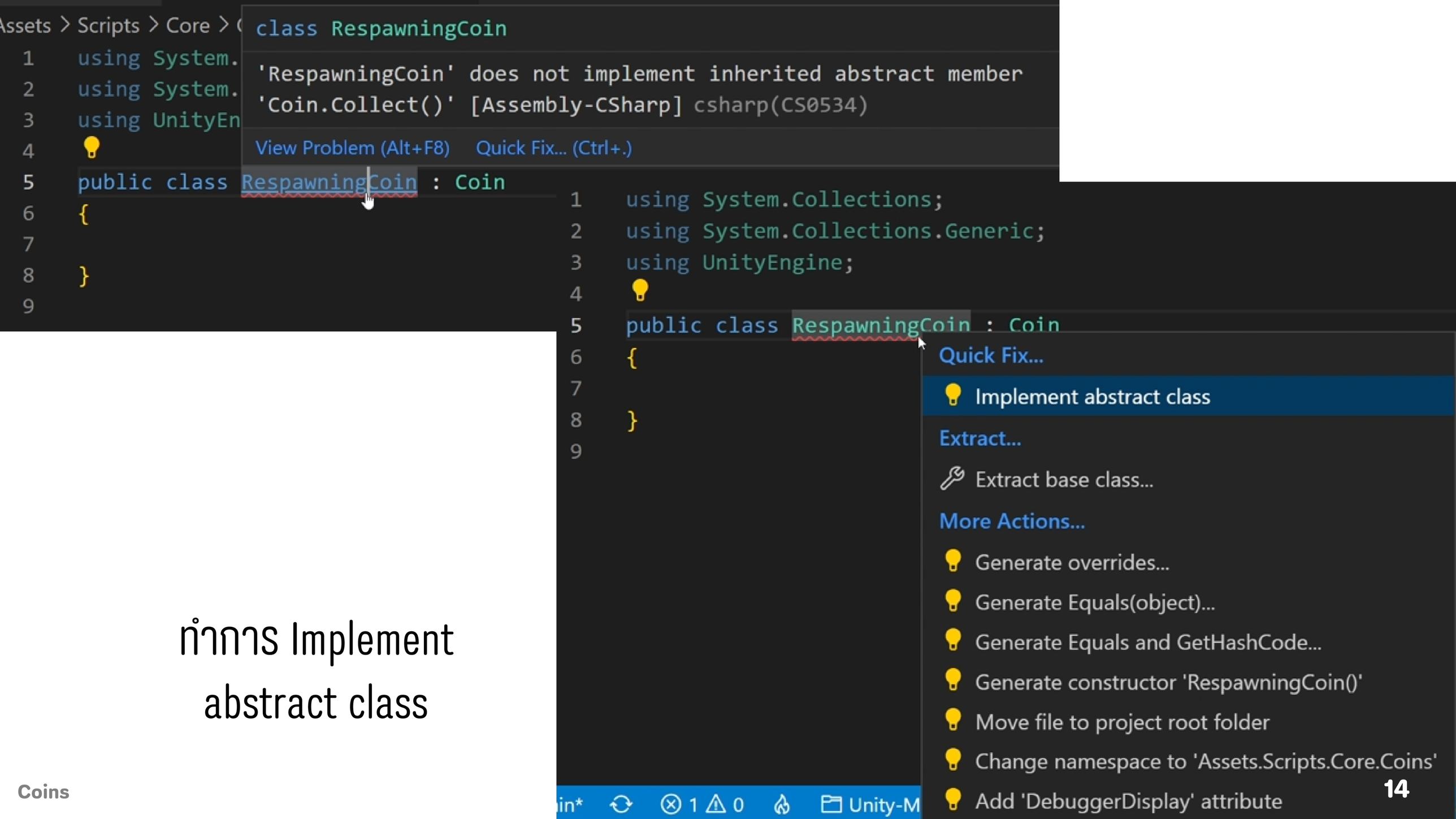
If we didn’t implement this then they’d both be able to pick up the coin and we don’t want duplicates like that.

เมื่อมีคนเก็บไปเราจะตั้งค่าให้เป็น true แล้วจะไม่มีใครสามารถหยิบได้อีก


Unity Script | 0 references
public abstract class Coin : MonoBehaviour
{
[SerializeField] private SpriteRenderer spriteRenderer;
protected int coinValue;
protected bool alreadyCollected;
0 references
public abstract int Collect();
0 references
public void SetValue(int value)
{
coinValue = value;
}
0 references
protected void Show(bool show)
{
spriteRenderer.enabled = show;
}
}



RespawningCoin



ทำการ Implement abstract class

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class RespawningCoin : Coin
{
    0 references
    public override int Collect()
    {

    }
}
```

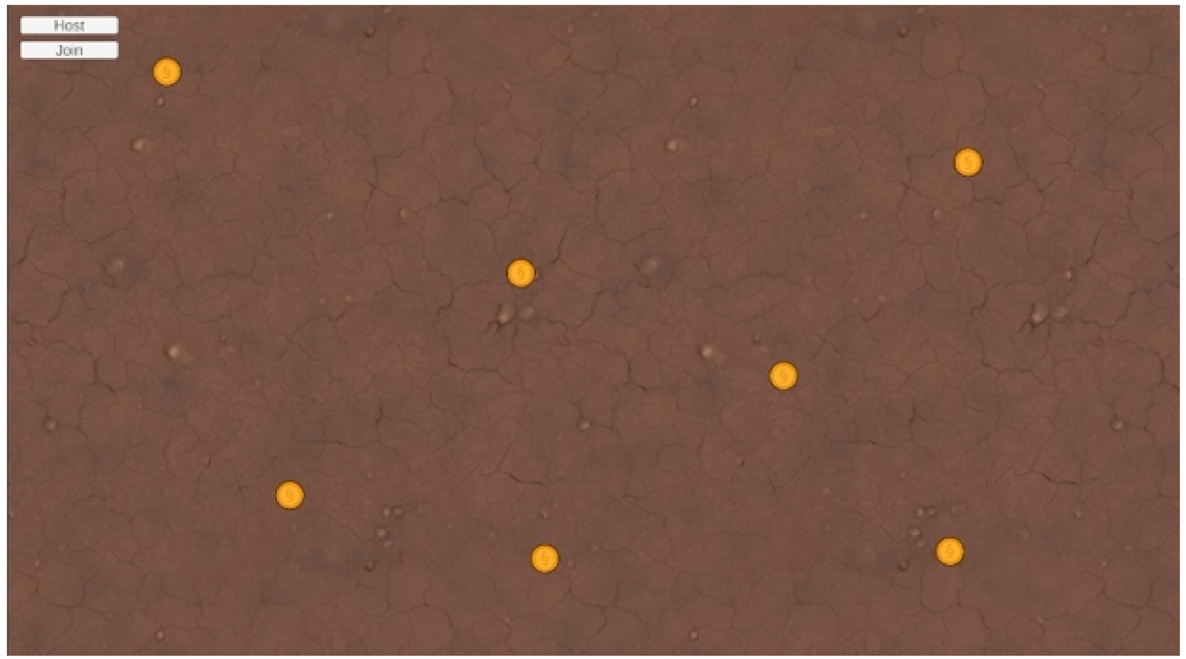
เปลี่ยน inherit ของ Coin
เป็น NetworkBehaviour

Assets > Scripts > Core > Coins > Coin.cs > Coin

```
using System.Collections;
using System.Collections.Generic;
using Unity.Netcode;
using UnityEngine;

1 reference
public abstract class Coin : NetworkBehaviour
{
    1 reference
    [SerializeField] private SpriteRenderer spriteRenderer;

    1 reference
    protected int coinValue;
    0 references
    protected bool alreadyCollected;

    0 references
    public abstract int Collect();
```

Collect Method

- Override the **Collect** method
- If we're not the server, then hide the coin and **return 0**
- If we've already collected the coin, **return 0**
- Otherwise set **alreadyCollected** to true
- Return the **coinValue**

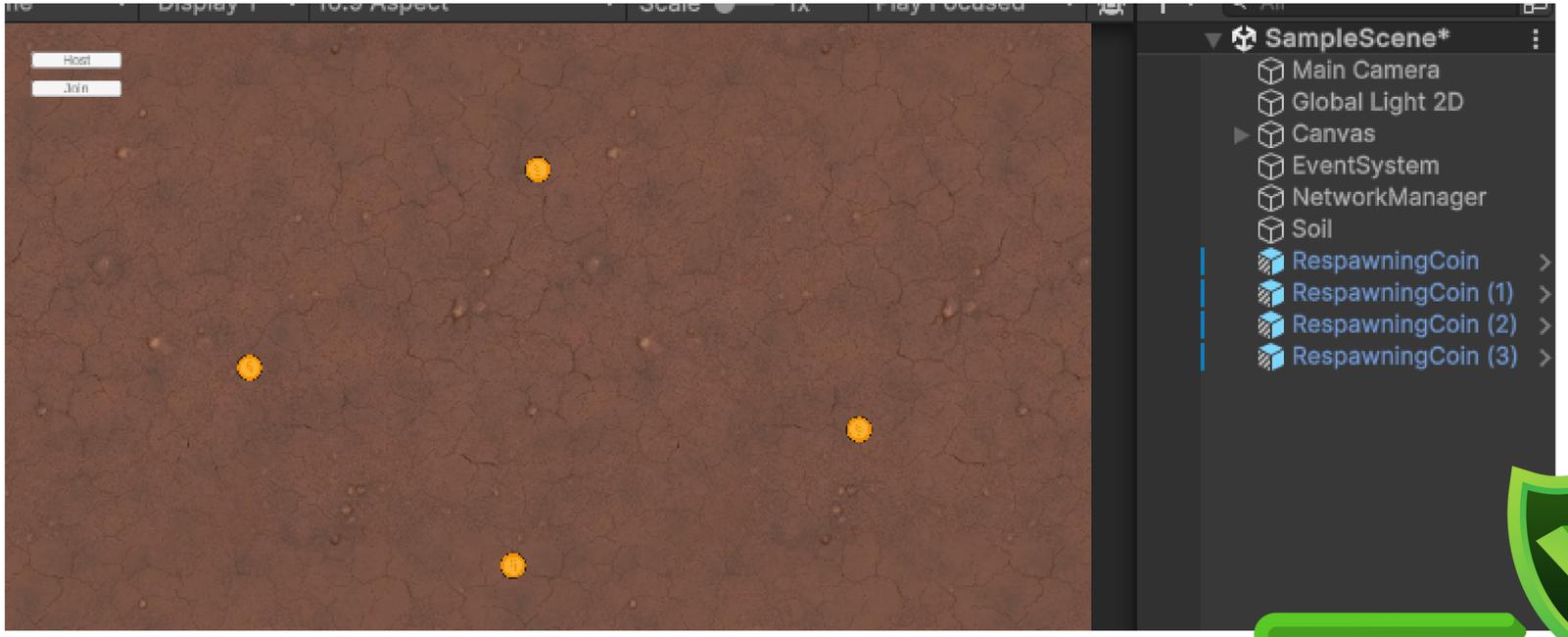


ทำ Function ให้สมบูรณ์

```
0 references
public override int Collect()
{
    if (!IsServer)
    {
        Show(false);
        return 0;
    }

    if (alreadyCollected) { return 0; }

    alreadyCollected = true;

    return coinValue;
}
}
```

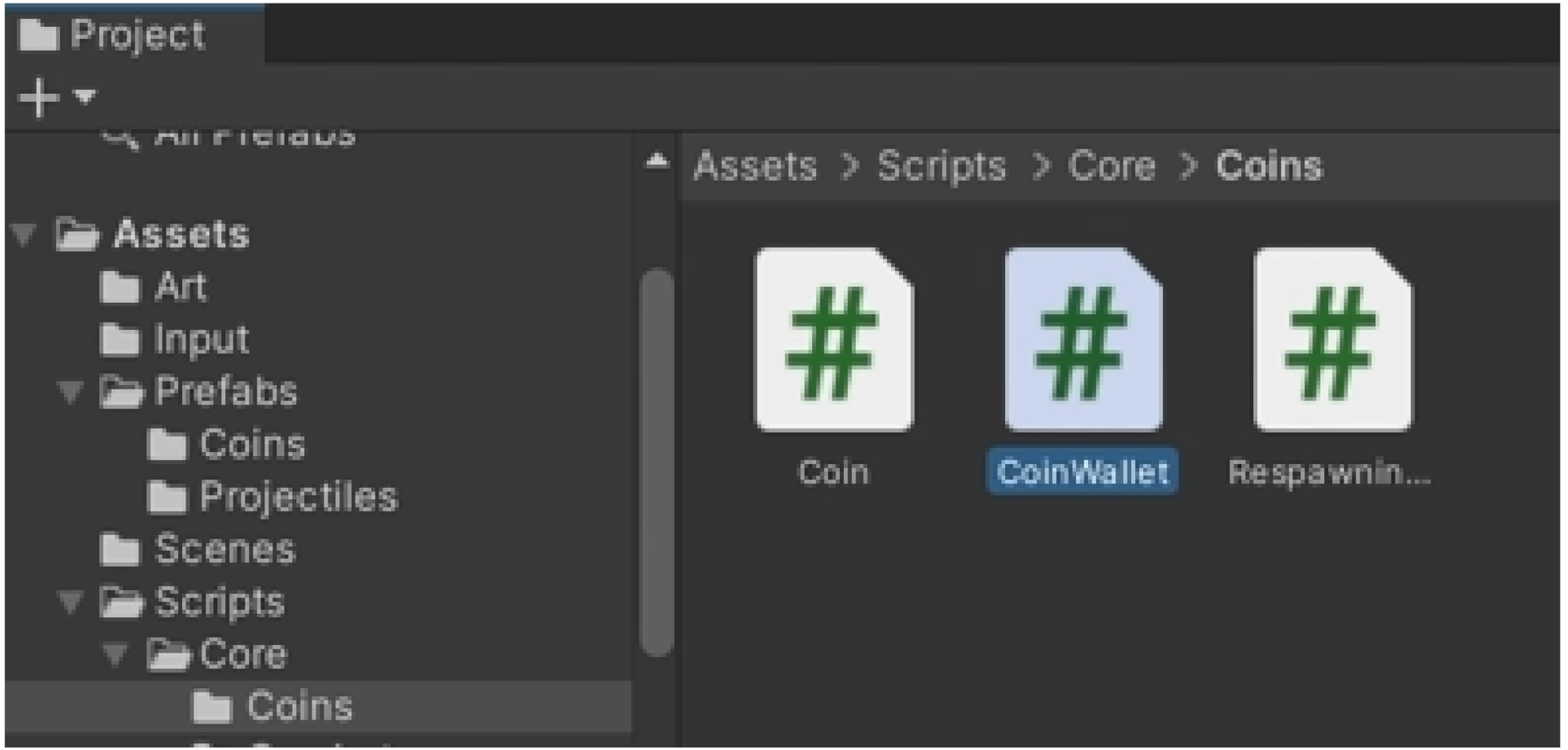

Congratulations!

Coins


Host
Join
SampleScene*
Main Camera
Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
RespawningCoin
RespawningCoin (1)
RespawningCoin (2)
RespawningCoin (3)


COMPLETED

Coins Wallet



สร้าง Script

CoinWallet

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
        Unity Script | 0 references
6   public class CoinWallet : NetworkBehaviour
7   {
8       public NetworkVariable<int> TotalCoins = new NetworkVariable<int>();
9   }
10
```

สร้างตัวแปร TotalCoins

Collect The Coins

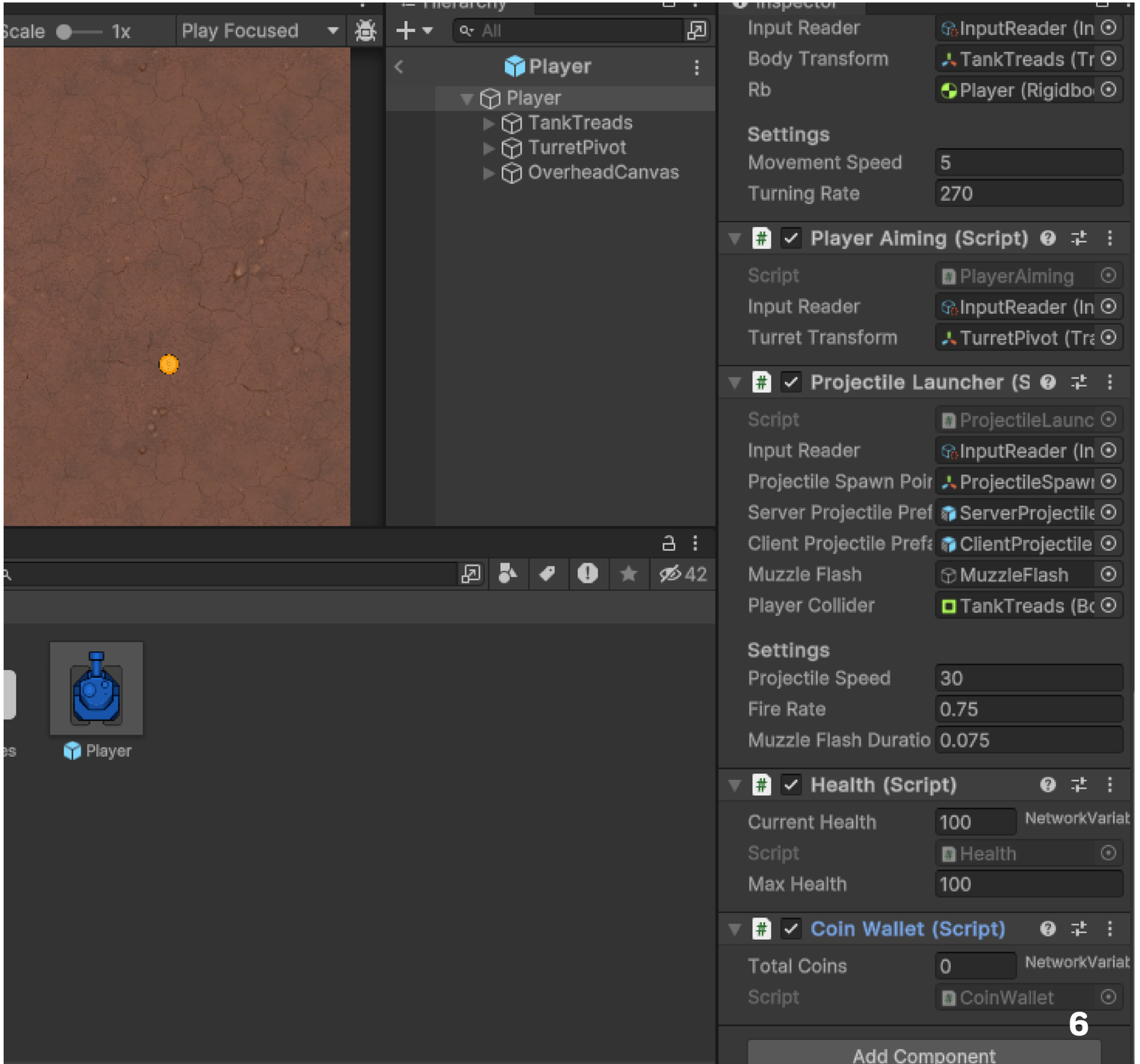
- Create **OnTriggerEnter2D(Collider2D)** method
- Try getting the **Coin** component off the collider
- If you successgully get the **Coin** component, call our **Collect()** method and store the **coinValue** in a temporary variable
- If we are not the server, **return**
- If we are the server, add to our **TotalCoins**

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
        Unity Script | 0 references
6   public class CoinWallet : NetworkBehaviour
7   {
8       public NetworkVariable<int> TotalCoins = new NetworkVariable<int>();
9
            Unity Message | 0 references
10      private void OnTriggerEnter2D(Collider2D col)
11      {
12          if(!col.TryGetComponent<Coin>(out Coin coin)) {  return; }
13
14          int coinValue = coin.Collect();
15
16          if(!IsServer) { return; }
17      }
18  }
19
```

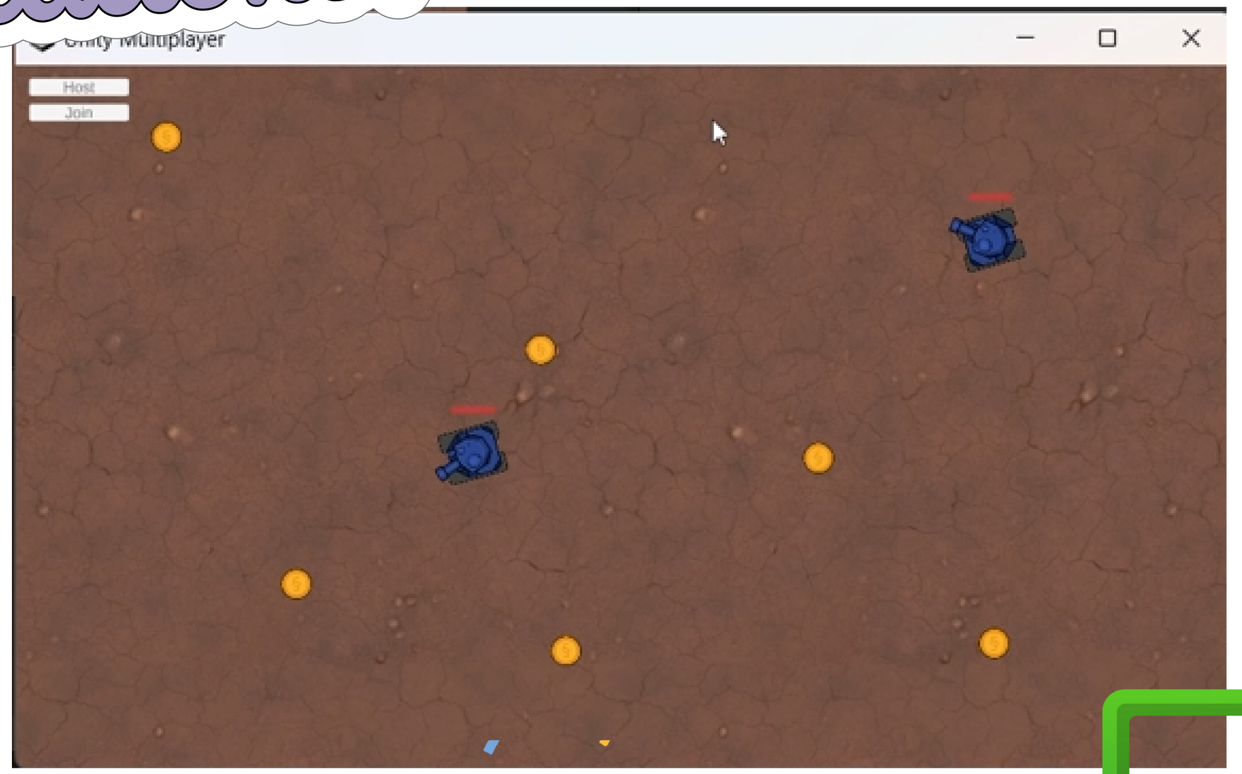
```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
    Unity Script | 0 references
6   public class CoinWallet : NetworkBehaviour
7   {
8       public NetworkVariable<int> TotalCoins = new NetworkVariable<int>();
9
        Unity Message | 0 references
10      private void OnTriggerEnter2D(Collider2D col)
11      {
12          if(!col.TryGetComponent<Coin>(out Coin coin)) {  return; }
13
14          int coinValue = coin.Collect();
15
16          if(!IsServer) { return; }
17
18          TotalCoins.Value += coinValue;
19      }
20  }
21
```

```
    Unity Script | 3 references
6   public abstract class Coin : NetworkBehaviour
7   {
8       [SerializeField] private SpriteRenderer spriteRenderer;
9
10      protected int coinValue = 10;
11      protected bool alreadyCollected;
12
            2 references
13      public abstract int Collect();
14
            0 references
15      public void SetValue(int value)
16      {
17          coinValue = value;
18      }
19
            1 reference
20      protected void Show(bool show)
21      {
22          spriteRenderer.enabled = show;
23      }
24  }
25
```

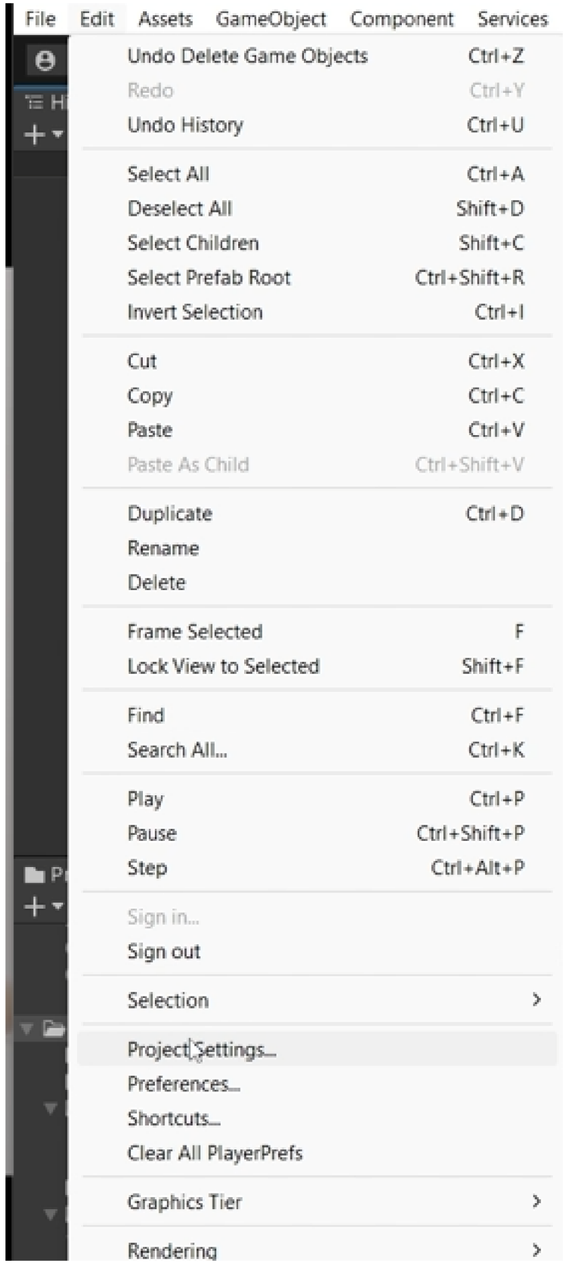
ใส่ Component ใน Player

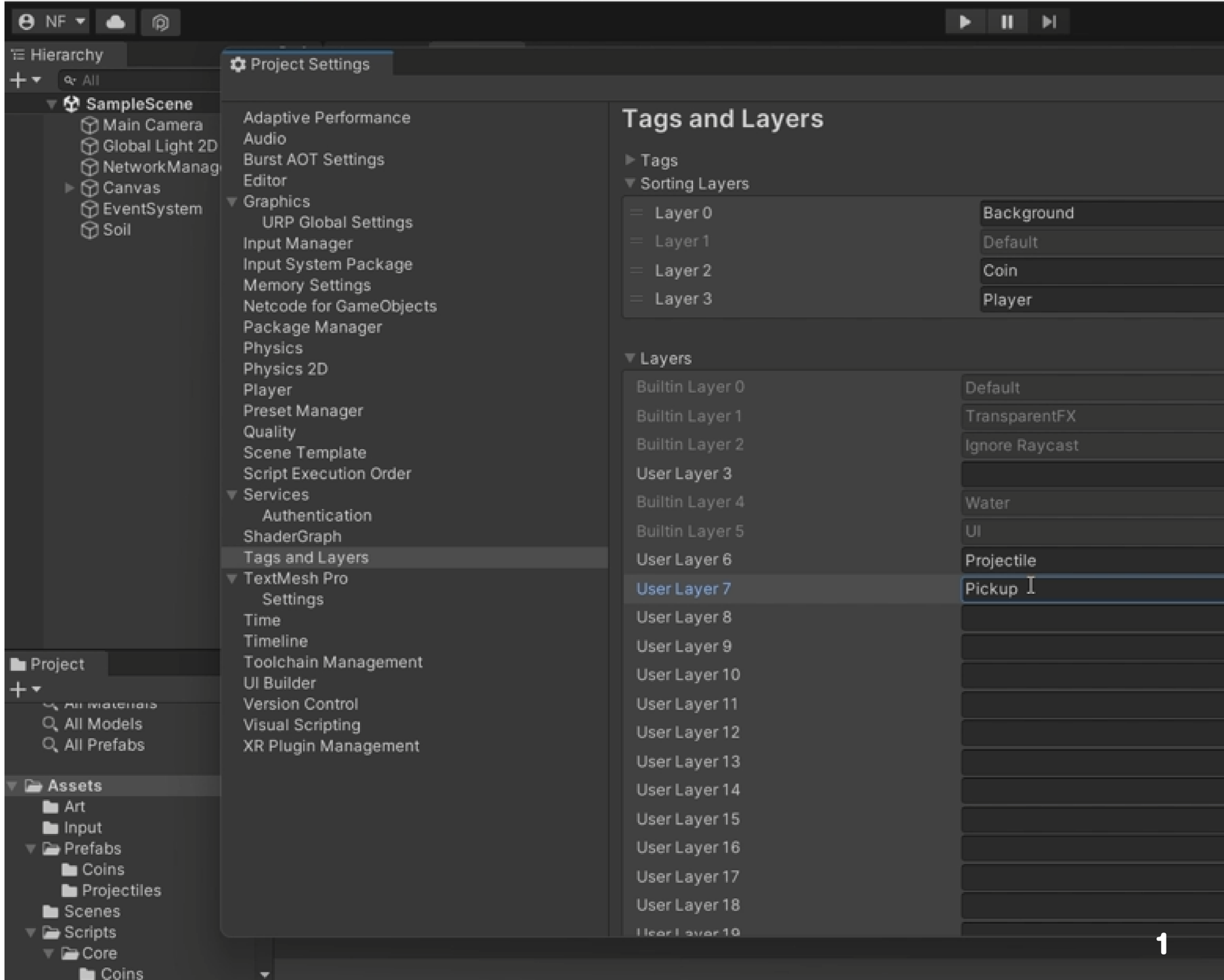


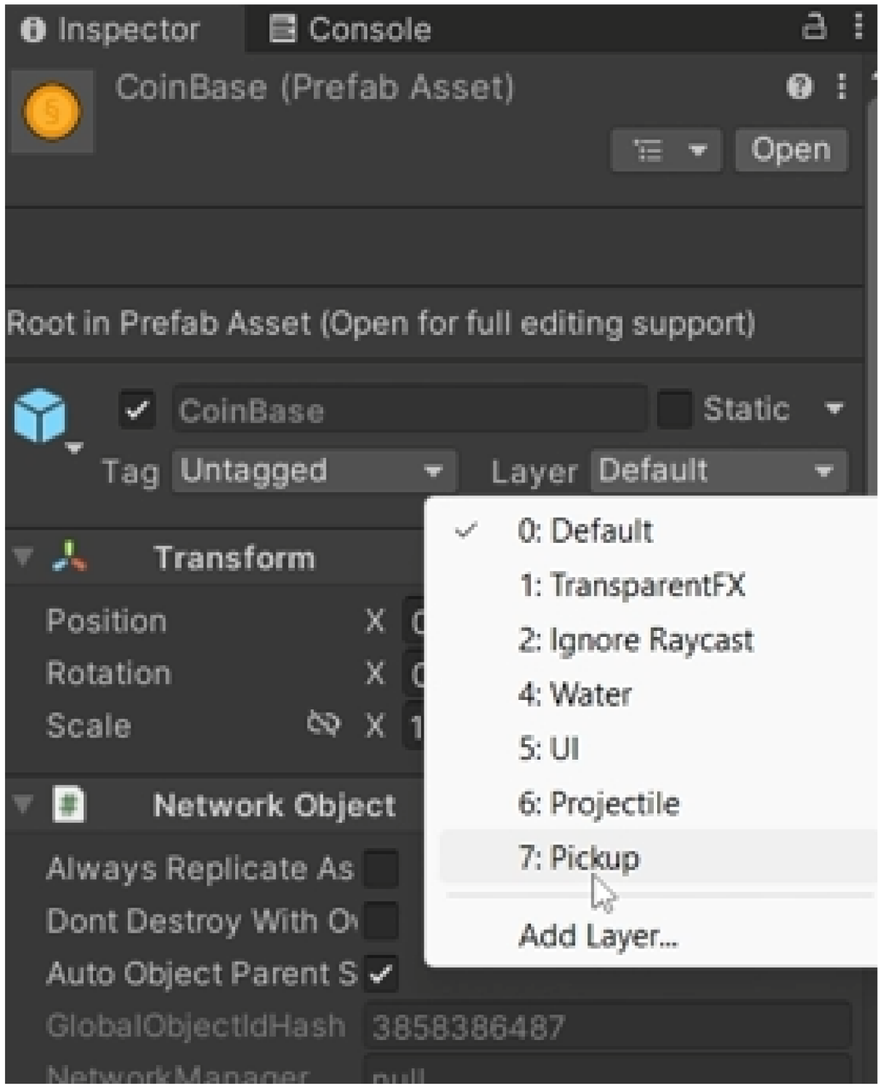
Coins Wallet

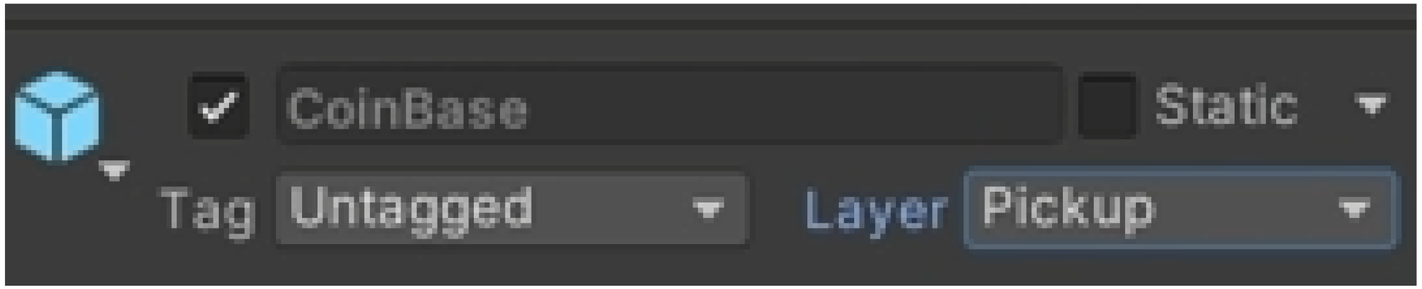


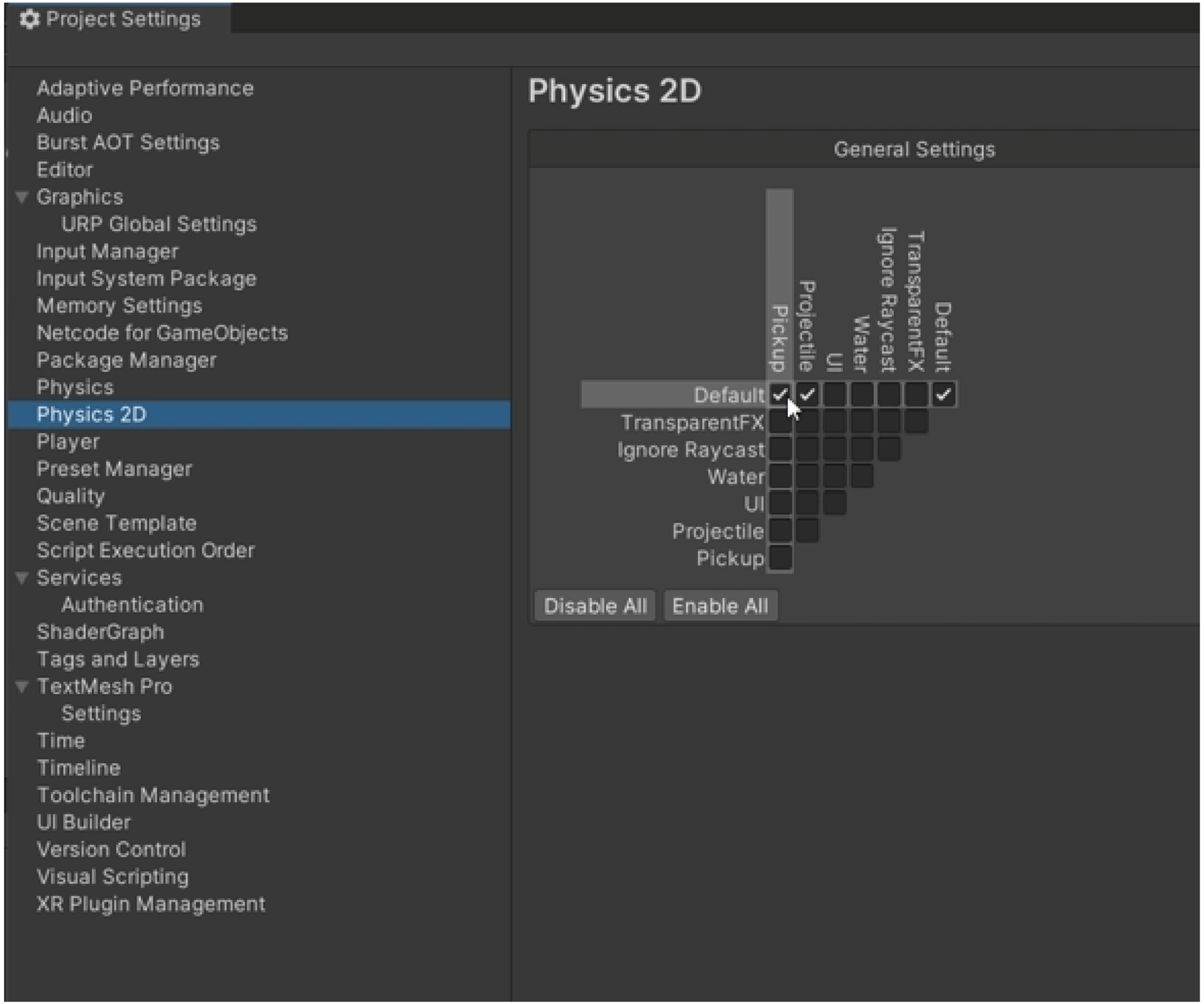
Coin Spawner


File
Edit
Assets
GameObject
Component
Services
Undo Delete Game Objects Ctrl+Z
Redo Ctrl+Y
Undo History Ctrl+U
Select All Ctrl+A
Deselect All Shift+D
Select Children Shift+C
Select Prefab Root Ctrl+Shift+R
Invert Selection Ctrl+I
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste As Child Ctrl+Shift+V
Duplicate Ctrl+D
Rename
Delete
Frame Selected F
Lock View to Selected Shift+F
Find Ctrl+F
Search All... Ctrl+K
Play Ctrl+P
Pause Ctrl+Shift+P
Step Ctrl+Alt+P
Sign in...
Sign out
Selection
Project Settings...
Preferences...
Shortcuts...
Clear All PlayerPrefs
Graphics Tier
Rendering


Hierarchy
SampleScene
Main Camera
Global Light 2D
NetworkManag
Canvas
EventSystem
Soil
Project Settings
Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
URP Global Settings
Input Manager
Input System Package
Memory Settings
Netcode for GameObjects
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Authentication
ShaderGraph
Tags and Layers
TextMesh Pro
Settings
Time
Timeline
Toolchain Management
UI Builder
Version Control
Visual Scripting
XR Plugin Management
Tags and Layers
Tags
Sorting Layers
Layer 0 Background
Layer 1 Default
Layer 2 Coin
Layer 3 Player
Layers
Builtin Layer 0 Default
Builtin Layer 1 TransparentFX
Builtin Layer 2 Ignore Raycast
User Layer 3
Builtin Layer 4 Water
Builtin Layer 5 UI
User Layer 6 Projectile
User Layer 7 Pickup
User Layer 8
User Layer 9
User Layer 10
User Layer 11
User Layer 12
User Layer 13
User Layer 14
User Layer 15
User Layer 16
User Layer 17
User Layer 18
Project
All Models
All Prefabs
Assets
Art
Input
Prefabs
Coins
Projectiles
Scenes
Scripts
Core
Coins

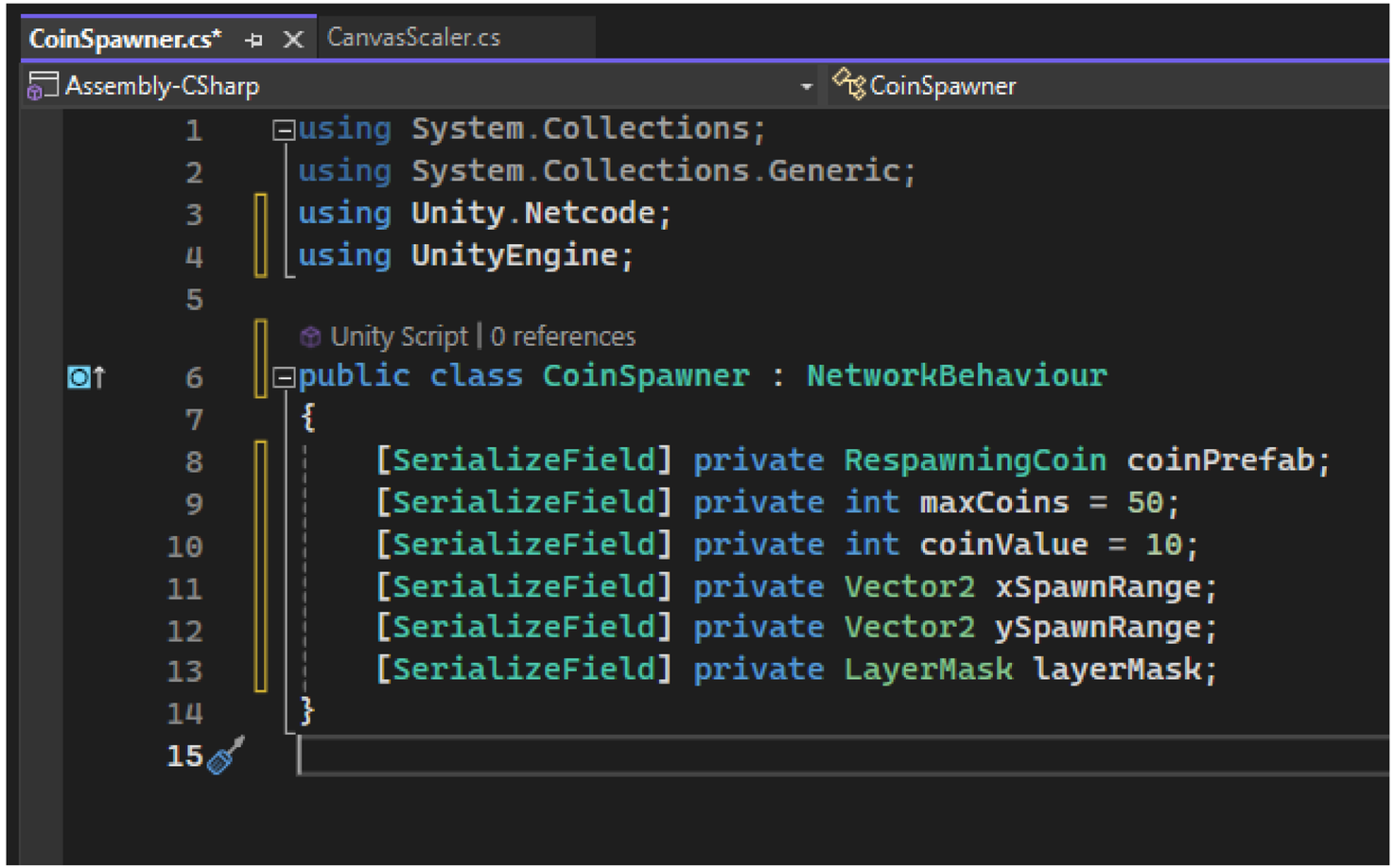

Inspector
Console
CoinBase (Prefab Asset)
Open
Root in Prefab Asset (Open for full editing support)
CoinBase
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Network Object
Always Replicate As
Dont Destroy With O
Auto Object Parent S
GlobalObjectIdHash
3858386487
0: Default
1: TransparentFX
2: Ignore Raycast
4: Water
5: UI
6: Projectile
7: Pickup
Add Layer...


CoinBase
Static
Tag
Untagged
Layer
Pickup

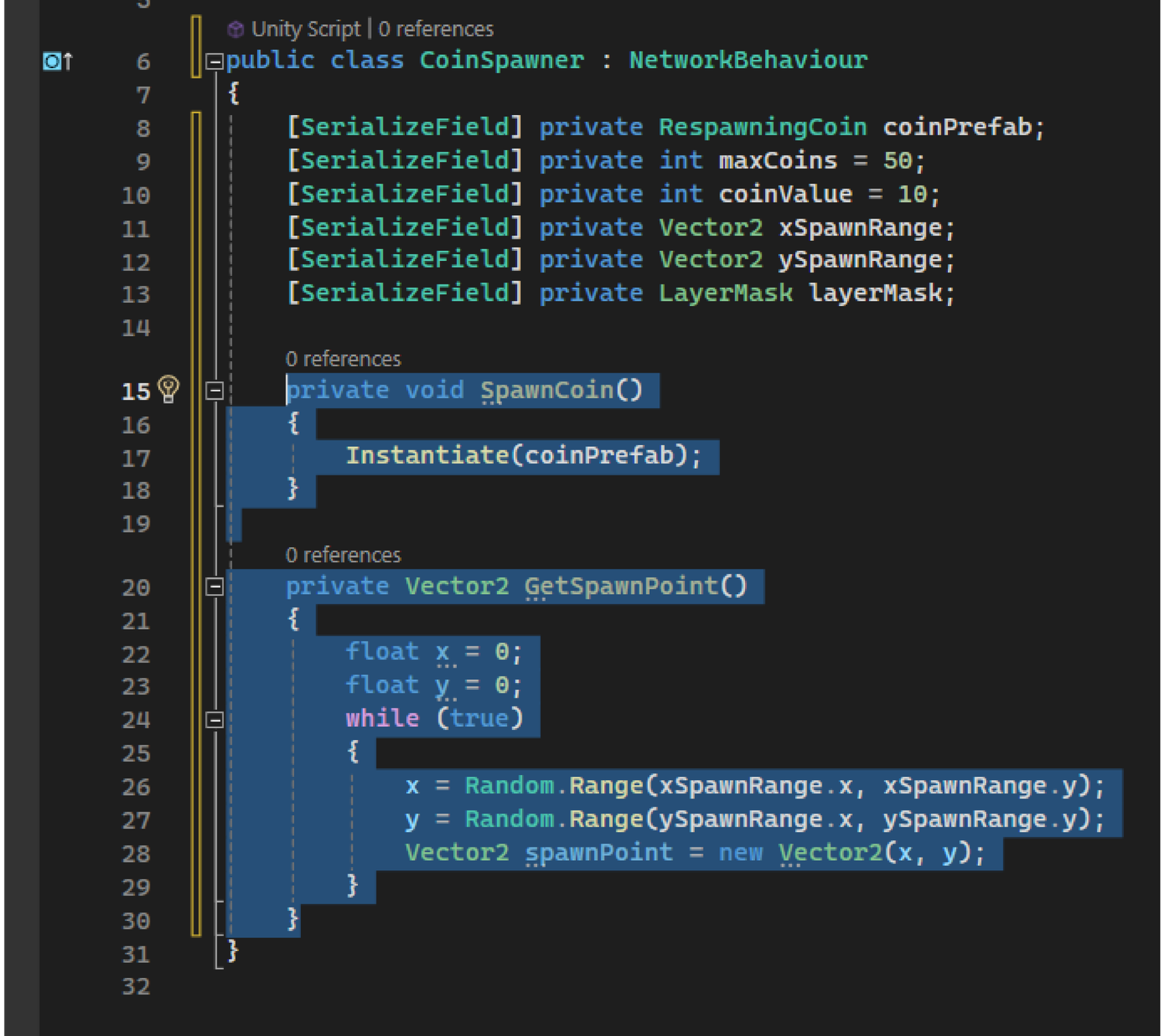

Project Settings
Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
URP Global Settings
Input Manager
Input System Package
Memory Settings
Netcode for GameObjects
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Authentication
ShaderGraph
Tags and Layers
TextMesh Pro
Settings
Time
Timeline
Toolchain Management
UI Builder
Version Control
Visual Scripting
XR Plugin Management
Physics 2D
General Settings
Pickup
Projectile
UI
Water
Ignore Raycast
TransparentFX
Default
Default
TransparentFX
Ignore Raycast
Water
UI
Projectile
Pickup
Disable All
Enable All

CoinSpawner

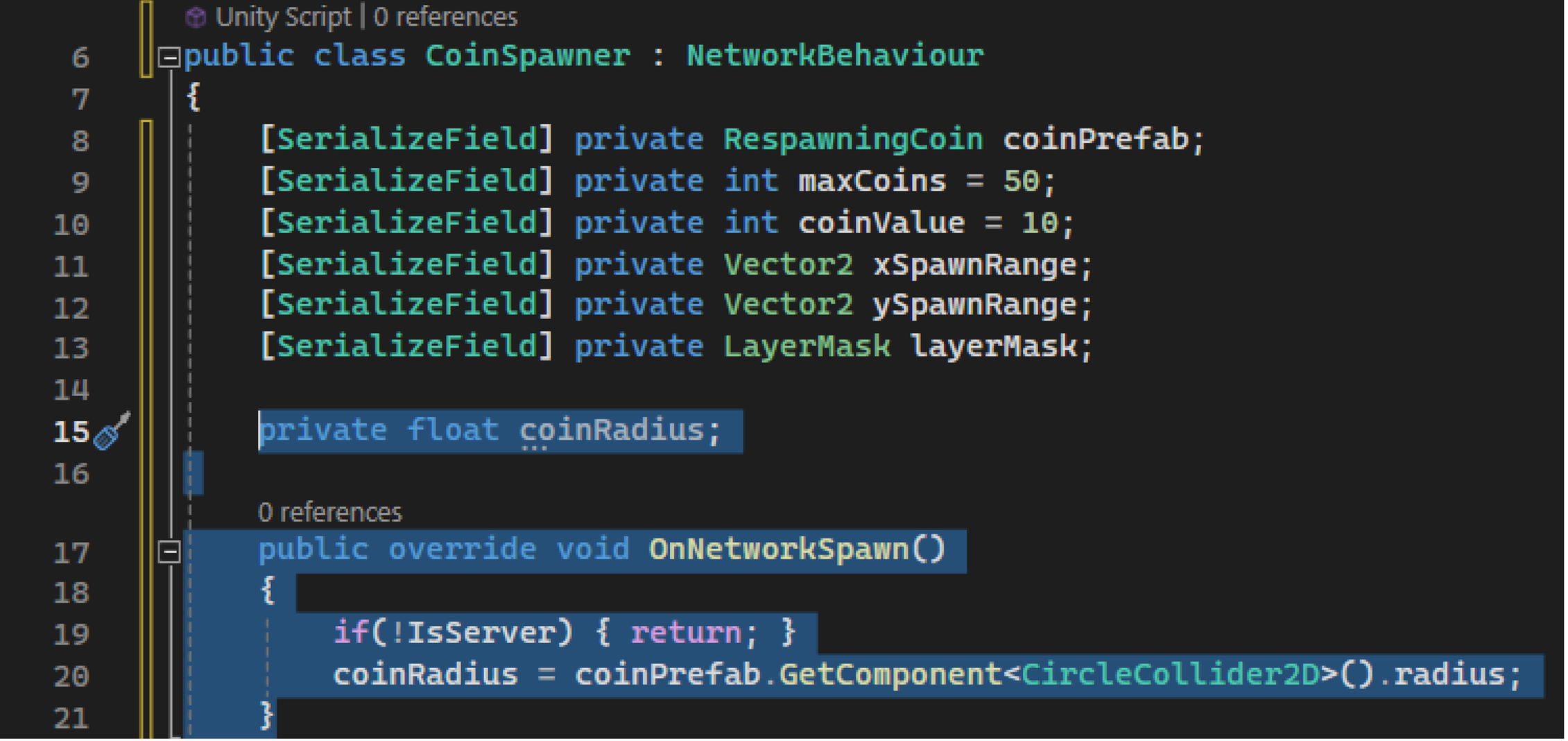
```
CoinSpawner.cs U
Assets > Scripts > Core > Coins > CoinSpawner.cs > CoinSpaw
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
    0 references
6   public class CoinSpawner : NetworkBehaviour
7   {
```



ใส่ตัวแปรให้ครบถ้วน



```
Unity Script | 0 references
public class CoinSpawner : NetworkBehaviour
{
    [SerializeField] private RespawningCoin coinPrefab;
    [SerializeField] private int maxCoins = 50;
    [SerializeField] private int coinValue = 10;
    [SerializeField] private Vector2 xSpawnRange;
    [SerializeField] private Vector2 ySpawnRange;
    [SerializeField] private LayerMask layerMask;

    0 references
    private void SpawnCoin()
    {
        Instantiate(coinPrefab);
    }

    0 references
    private Vector2 GetSpawnPoint()
    {
        float x = 0;
        float y = 0;
        while (true)
        {
            x = Random.Range(xSpawnRange.x, xSpawnRange.y);
            y = Random.Range(ySpawnRange.x, ySpawnRange.y);
            Vector2 spawnPoint = new Vector2(x, y);
        }
    }
}
```


Unity Script | 0 references
public class CoinSpawner : NetworkBehaviour
{
[SerializeField] private RespawningCoin coinPrefab;
[SerializeField] private int maxCoins = 50;
[SerializeField] private int coinValue = 10;
[SerializeField] private Vector2 xSpawnRange;
[SerializeField] private Vector2 ySpawnRange;
[SerializeField] private LayerMask layerMask;
private float coinRadius;
0 references
public override void OnNetworkSpawn()
{
if(!IsServer) { return; }
coinRadius = coinPrefab.GetComponent<CircleCollider2D>().radius;
}

```
Unity Script | 0 references
public class CoinSpawner : NetworkBehaviour
{
    [SerializeField] private RespawningCoin coinPrefab;
    [SerializeField] private int maxCoins = 50;
    [SerializeField] private int coinValue = 10;
    [SerializeField] private Vector2 xSpawnRange;
    [SerializeField] private Vector2 ySpawnRange;
    [SerializeField] private LayerMask layerMask;

    private Collider2D[] coinBuffer = new Collider2D[1];

    private float coinRadius;
```

```
1 reference
private Vector2 GetSpawnPoint()
{
    float x = 0;
    float y = 0;
    while (true)
    {
        x = Random.Range(xSpawnRange.x, xSpawnRange.y);
        y = Random.Range(ySpawnRange.x, ySpawnRange.y);
        Vector2 spawnPoint = new Vector2(x, y);
        ContactFilter2D contactFilter2D = new ContactFilter2D();
        contactFilter2D.layerMask = layerMask;
        int numColliders = Physics2D.OverlapCircle(spawnPoint, coinRadius, contactFilter2D, coinBuffer);
        if(numColliders == 0)
        {
            return spawnPoint;
        }

    }
}
```

```
1 reference
private void SpawnCoin()
{
    RespawningCoin coinInstance = Instantiate(
        coinPrefab,
        GetSpawnPoint(),
        Quaternion.identity);

    coinInstance.SetValue(coinValue);
    coinInstance.GetComponent<NetworkObject>().Spawn();

}
```

```
public override void OnNetworkSpawn()
{
    if(!IsServer) { return; }
    coinRadius = coinPrefab.GetComponent<CircleCollider2D>().radius;

    for(int i = 0;i<maxCoins;i++)
    {
        SpawnCoin();
    }
}
```

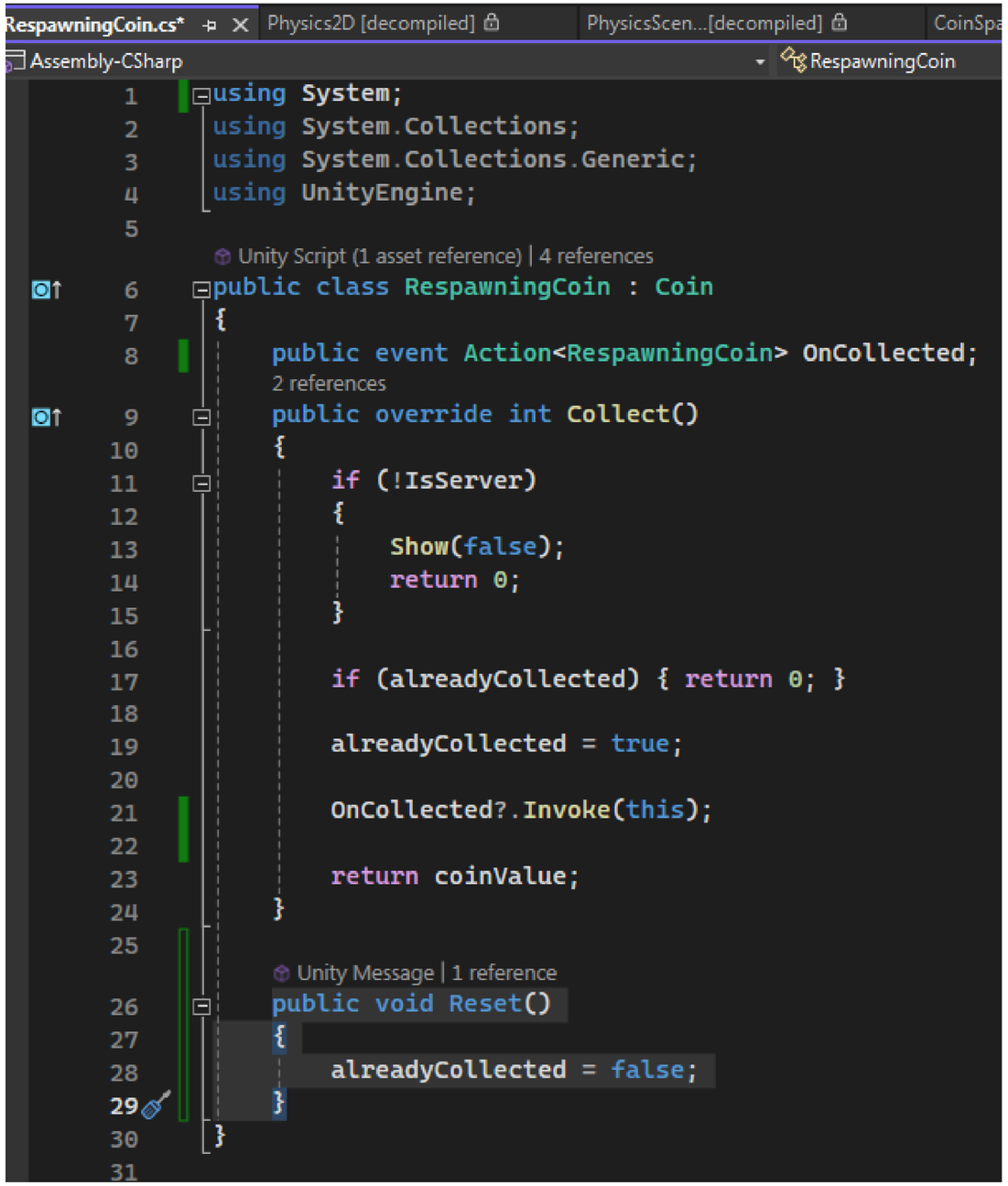
RespawningCoin.cs* | Physics2D [decompiled] | PhysicsScen...[decompiled] | CoinSpawner.

Assembly-CSharp | RespawningCoin

```
1   using System;
2   using System.Collections;
3   using System.Collections.Generic;
4   using UnityEngine;
5
    Unity Script (1 asset reference) | 3 references
6   public class RespawningCoin : Coin
7   {
8       public event Action<RespawningCoin> OnCollected;
        2 references
9       public override int Collect()
10      {
11          if (!IsServer)
12          {
13              Show(false);
14              return 0;
15          }
16
17          if (alreadyCollected) { return 0; }
18
19          alreadyCollected = true;
20
21          return coinValue;
22      }
23  }
```

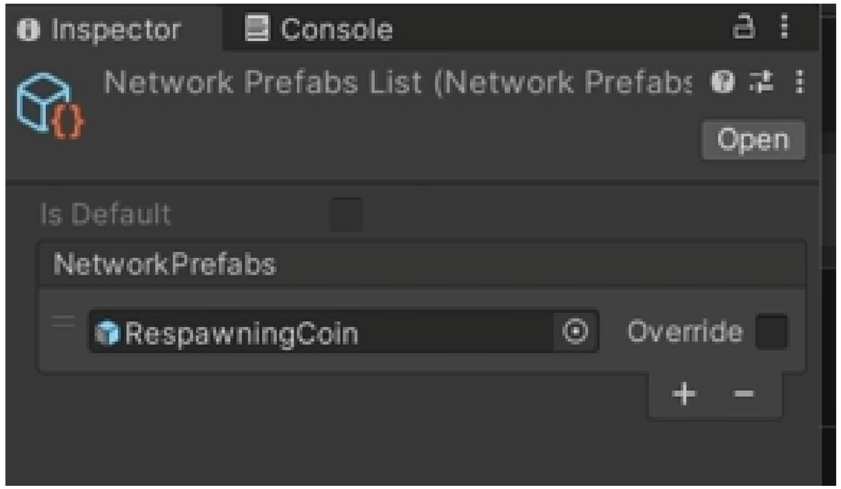
```
public override int Collect()
{
    if (!IsServer)
    {
        Show(false);
        return 0;
    }

    if (alreadyCollected) { return 0; }

    alreadyCollected = true;

    OnCollected?.Invoke(this);

    return coinValue;
}
```

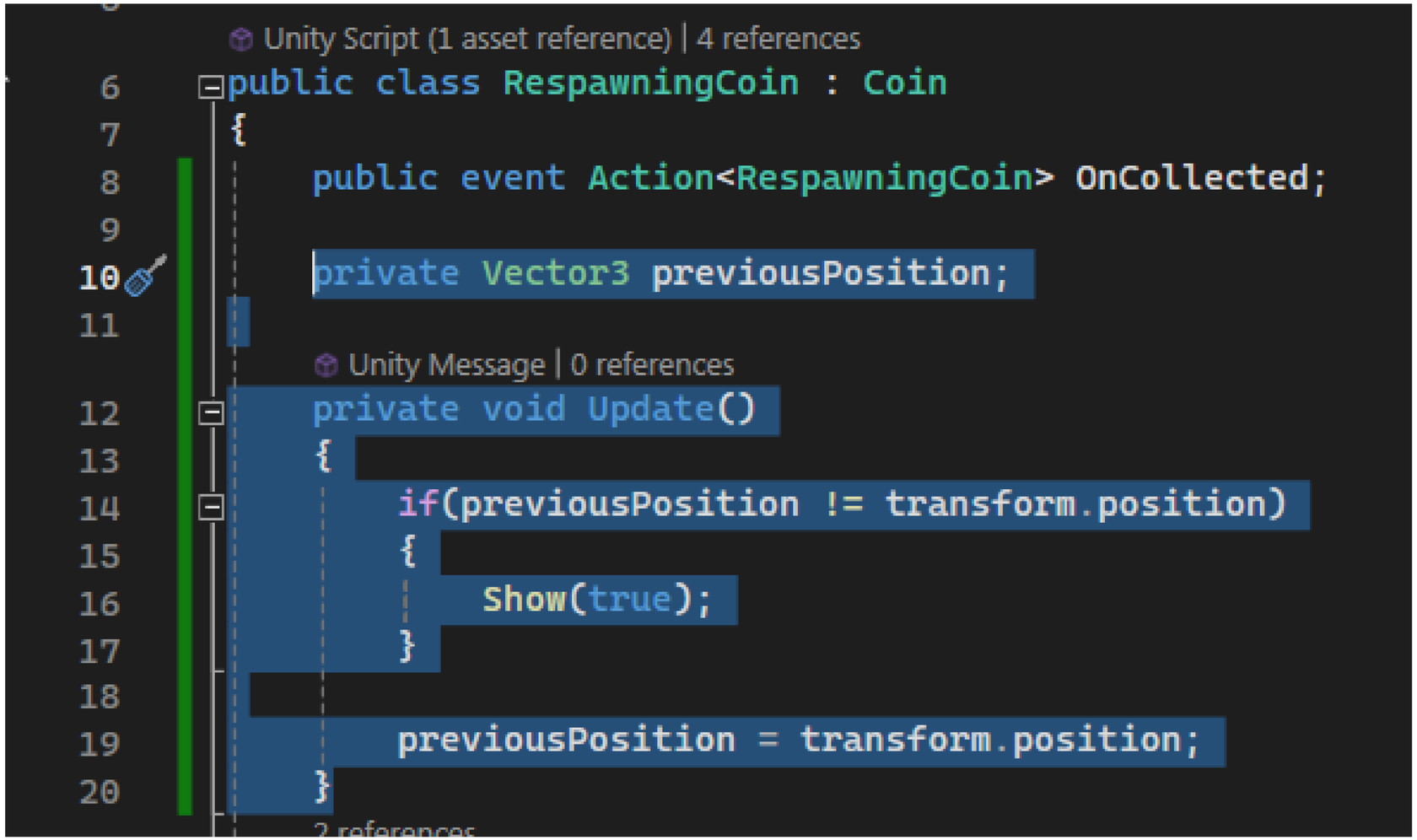
```
2 references
private void SpawnCoin()
{
    RespawningCoin coinInstance = Instantiate(
        coinPrefab,
        GetSpawnPoint(),
        Quaternion.identity);

    coinInstance.SetValue(coinValue);
    coinInstance.GetComponent<NetworkObject>().Spawn();

    coinInstance.OnCollected += HandleCoinCollected;
}
1 reference
private void HandleCoinCollected(RespawningCoin coin)
{
    coin.transform.position = GetSpawnPoint();
    coin.Reset();
}
```

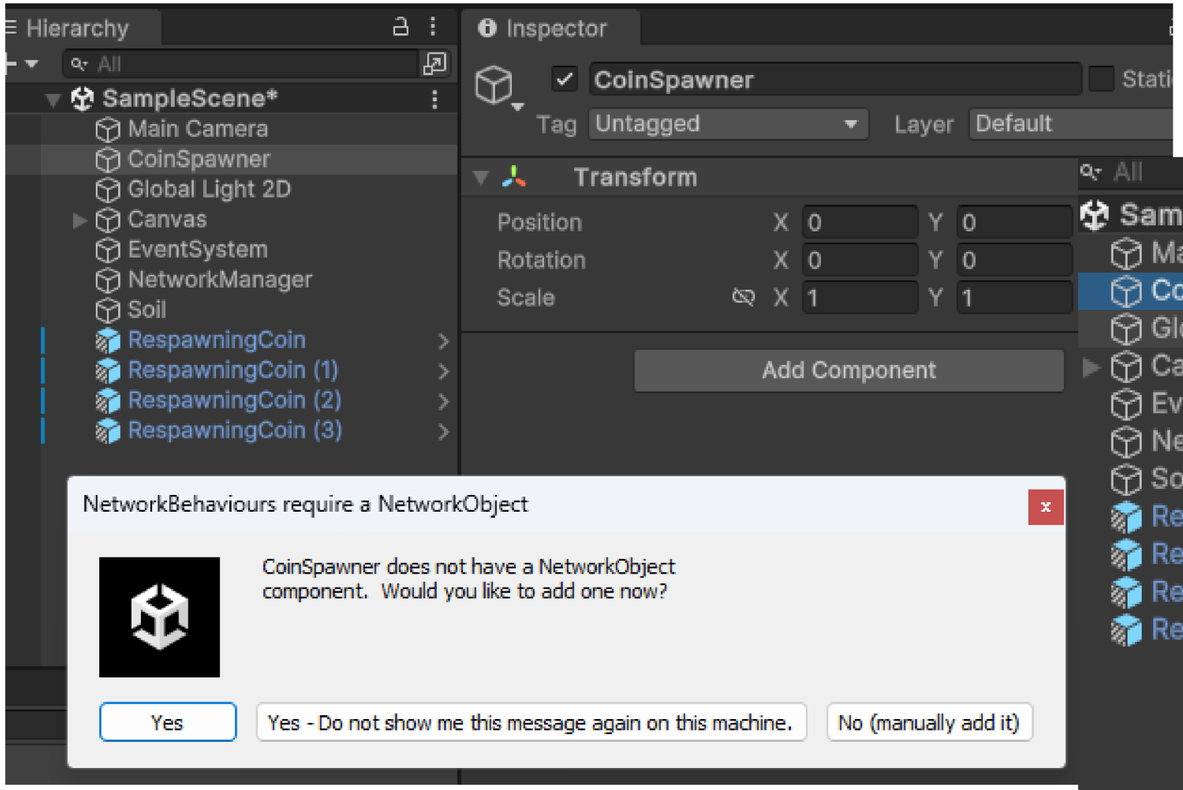

RespawningCoin.cs*
Physics2D [decompiled]
PhysicsScen...[decompiled]
CoinSpa
Assembly-CSharp
RespawningCoin
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
Unity Script (1 asset reference) | 4 references
public class RespawningCoin : Coin
{
public event Action<RespawningCoin> OnCollected;
2 references
public override int Collect()
{
if (!IsServer)
{
Show(false);
return 0;
}
if (alreadyCollected) { return 0; }
alreadyCollected = true;
OnCollected?.Invoke(this);
return coinValue;
}
Unity Message | 1 reference
public void Reset()
{
alreadyCollected = false;
}
}

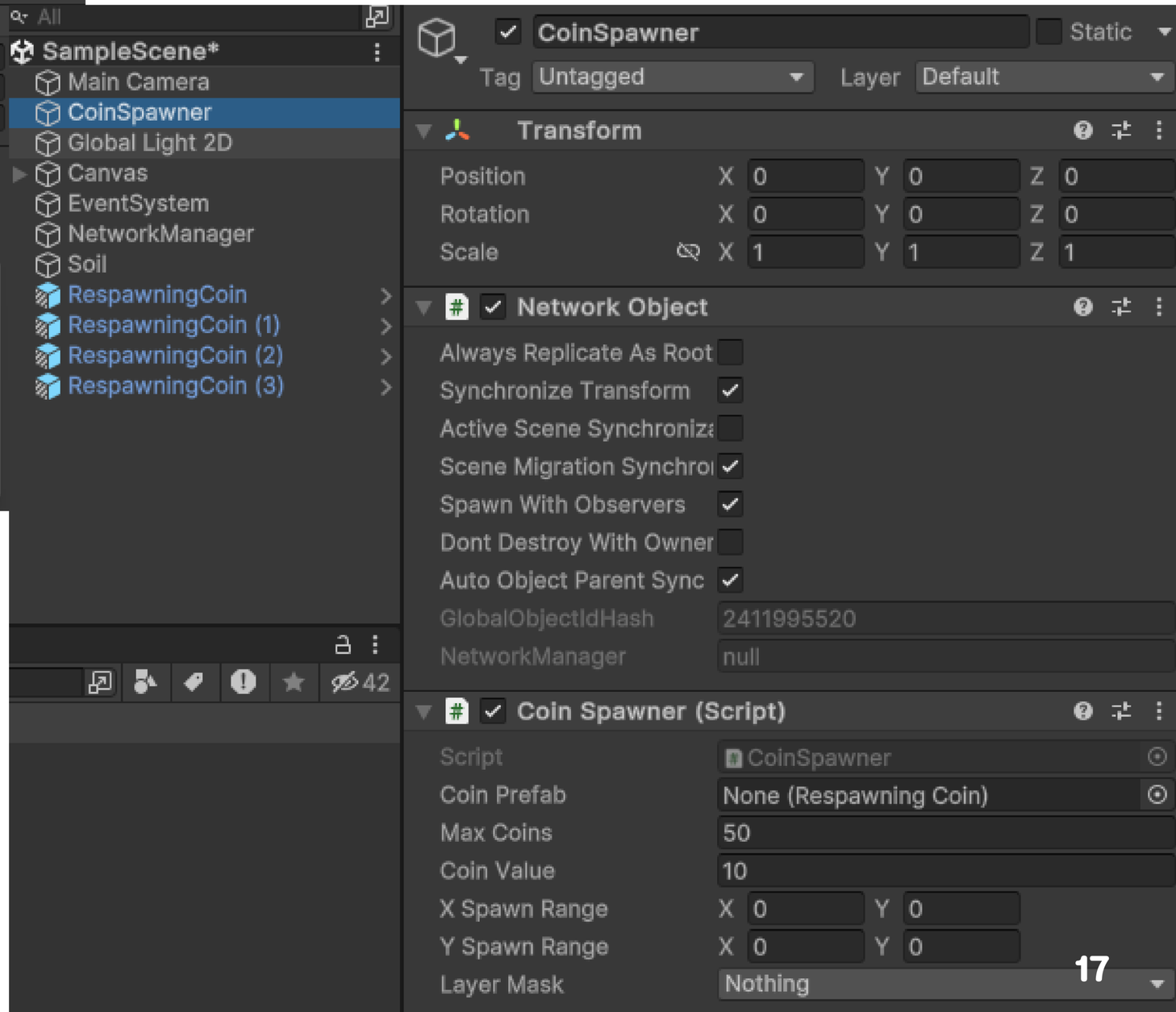
Respawning Coin

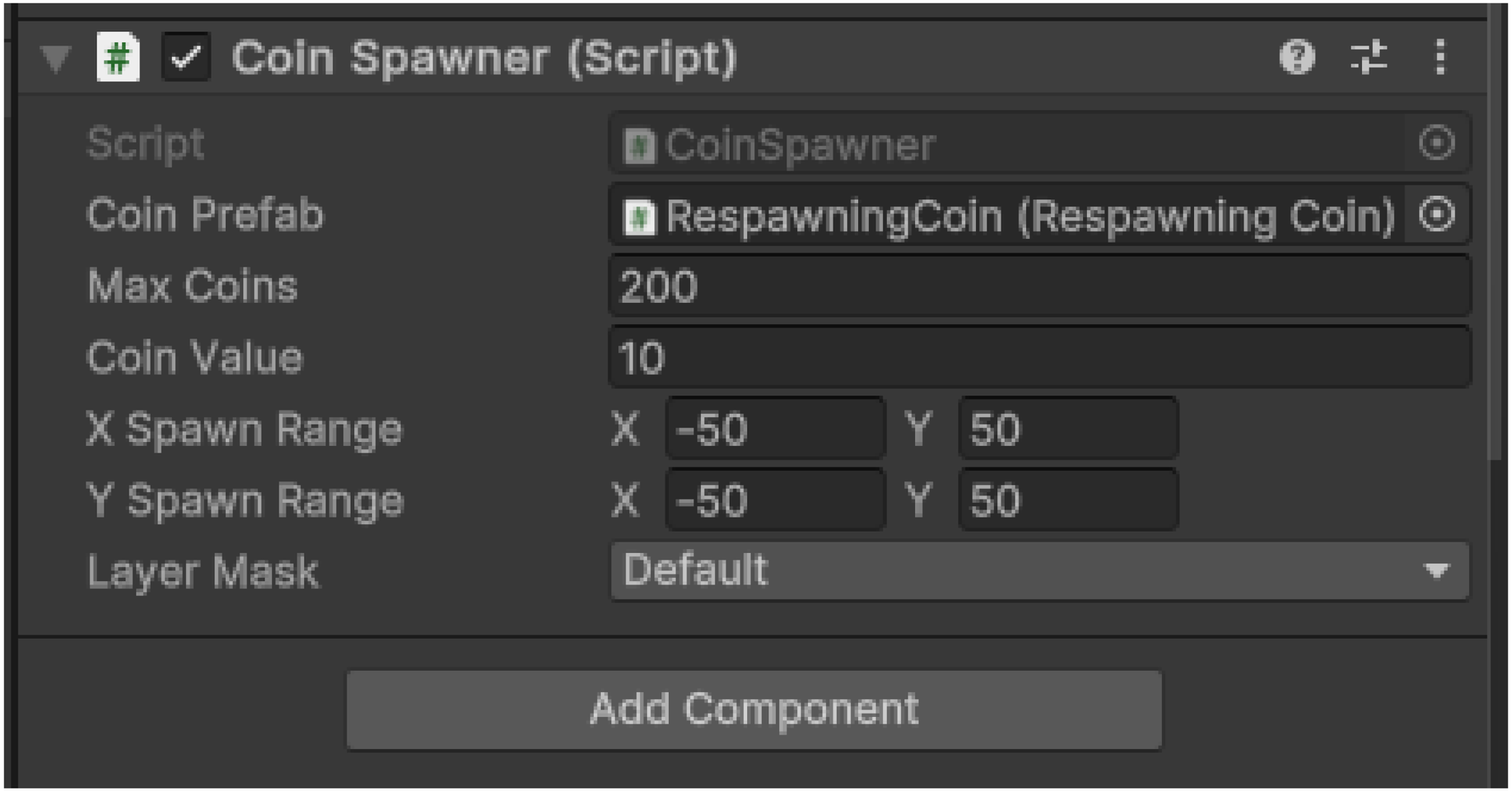
- Store the coins **Vector3 previousPosition**
- Check each frame to see if the coin has moved
- If it has moved, show the coin again
- Update the **previousPosition**
- This code only needs to be executed for non-server players

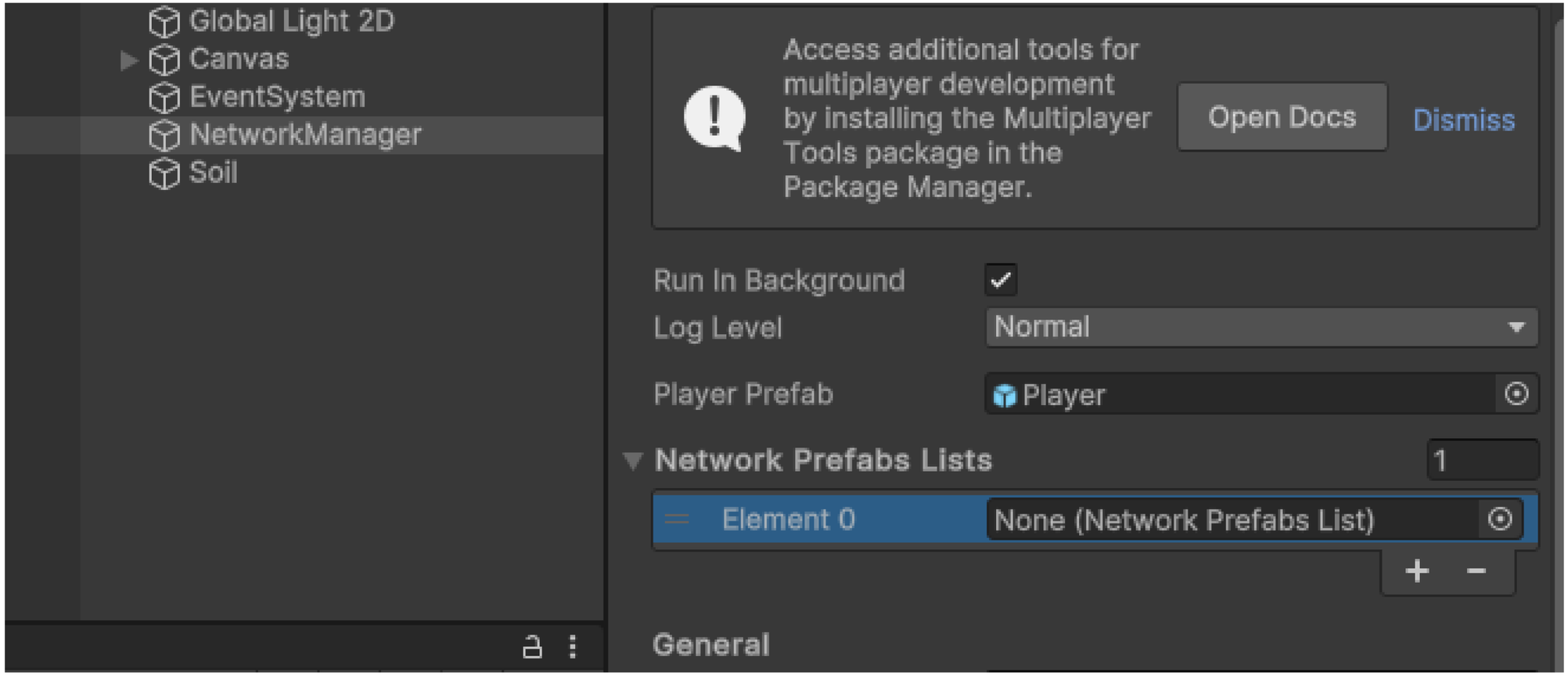


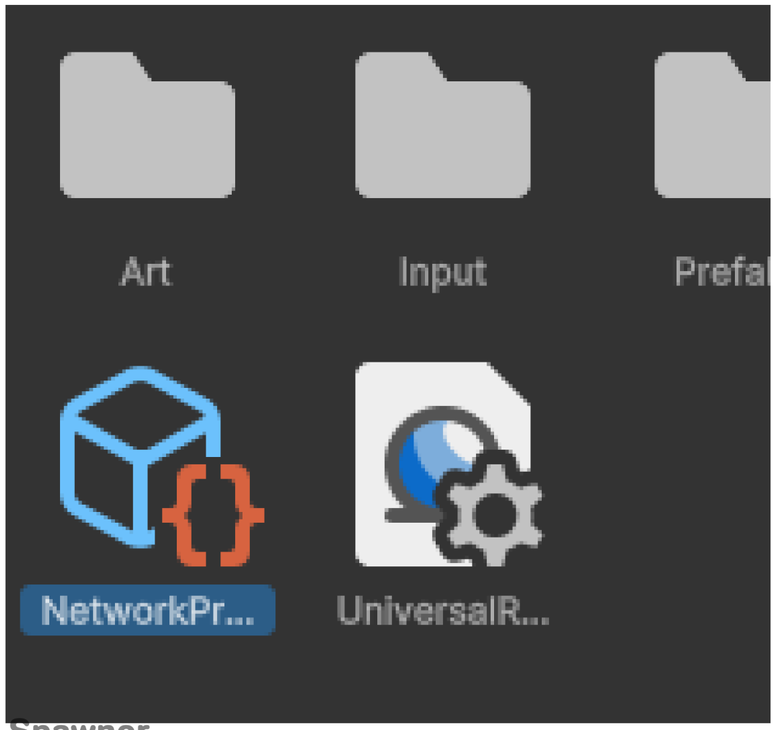

Unity Script (1 asset reference) | 4 references
public class RespawningCoin : Coin
{
public event Action<RespawningCoin> OnCollected;
private Vector3 previousPosition;
Unity Message | 0 references
private void Update()
{
if(previousPosition != transform.position)
{
Show(true);
}
previousPosition = transform.position;
}


Hierarchy
All
SampleScene*
Main Camera
CoinSpawner
Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
RespawningCoin
RespawningCoin (1)
RespawningCoin (2)
RespawningCoin (3)
Inspector
CoinSpawner
Tag Untagged
Layer Default
Transform
Position X 0 Y 0
Rotation X 0 Y 0
Scale X 1 Y 1
Add Component
NetworkBehaviours require a NetworkObject
CoinSpawner does not have a NetworkObject component. Would you like to add one now?
Yes
Yes - Do not show me this message again on this machine.
No (manually add it)


All
SampleScene*
Main Camera
CoinSpawner
Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
RespawningCoin
RespawningCoin (1)
RespawningCoin (2)
RespawningCoin (3)
42
CoinSpawner
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Network Object
Always Replicate As Root
Synchronize Transform
Active Scene Synchroniza
Scene Migration Synchroi
Spawn With Observers
Dont Destroy With Owner
Auto Object Parent Sync
GlobalObjectIdHash 2411995520
NetworkManager null
Coin Spawner (Script)
Script CoinSpawner
Coin Prefab None (Respawning Coin)
Max Coins 50
Coin Value 10
X Spawn Range X 0 Y 0
Y Spawn Range X 0 Y 0
Layer Mask Nothing


Coin Spawner (Script)
Script
CoinSpawner
Coin Prefab
RespawningCoin (Respawning Coin)
Max Coins
200
Coin Value
10
X Spawn Range
X -50
Y 50
Y Spawn Range
X -50
Y 50
Layer Mask
Default
Add Component


Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
Access additional tools for multiplayer development by installing the Multiplayer Tools package in the Package Manager.
Open Docs
Dismiss
Run In Background
Log Level
Normal
Player Prefab
Player
Network Prefabs Lists
1
Element 0
None (Network Prefabs List)
General

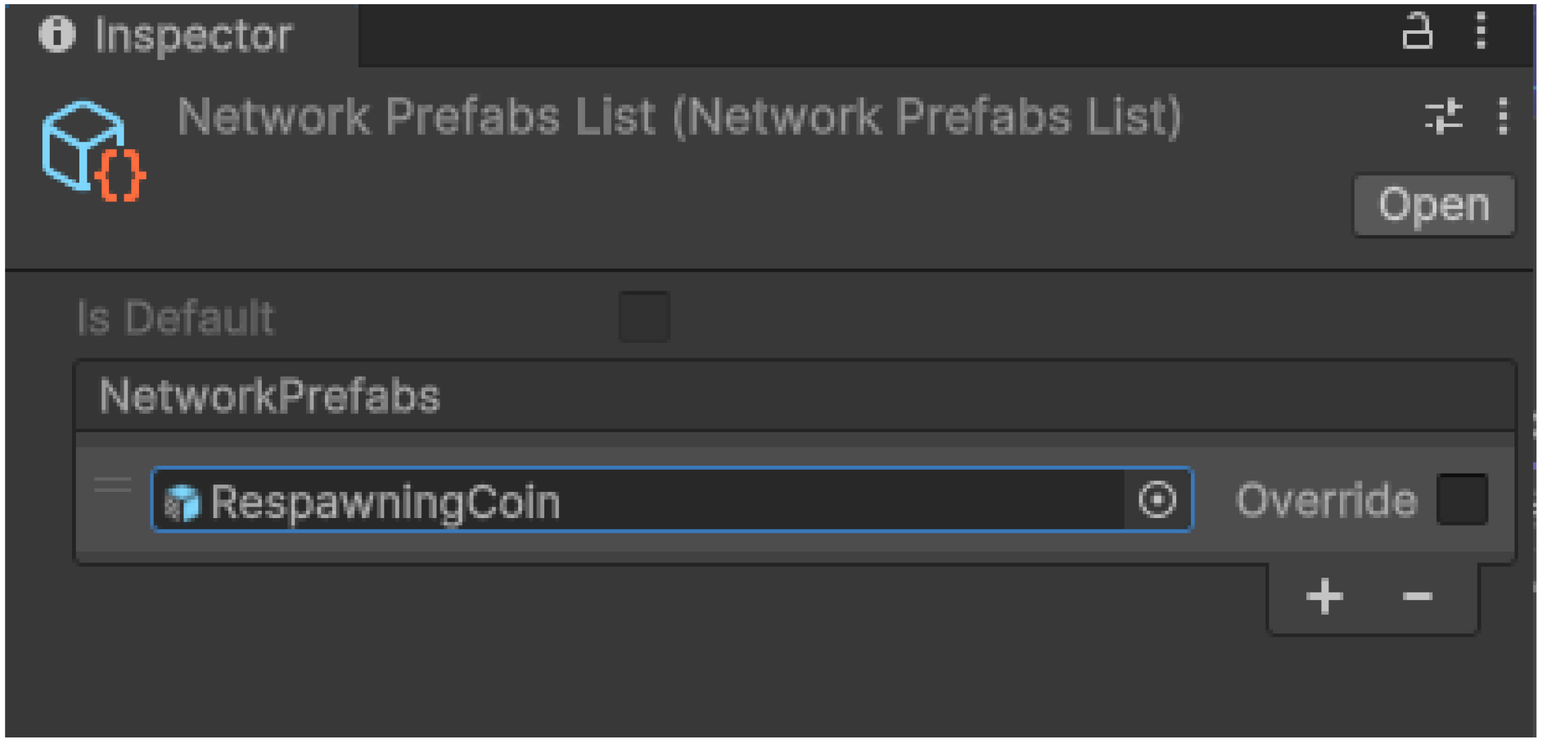


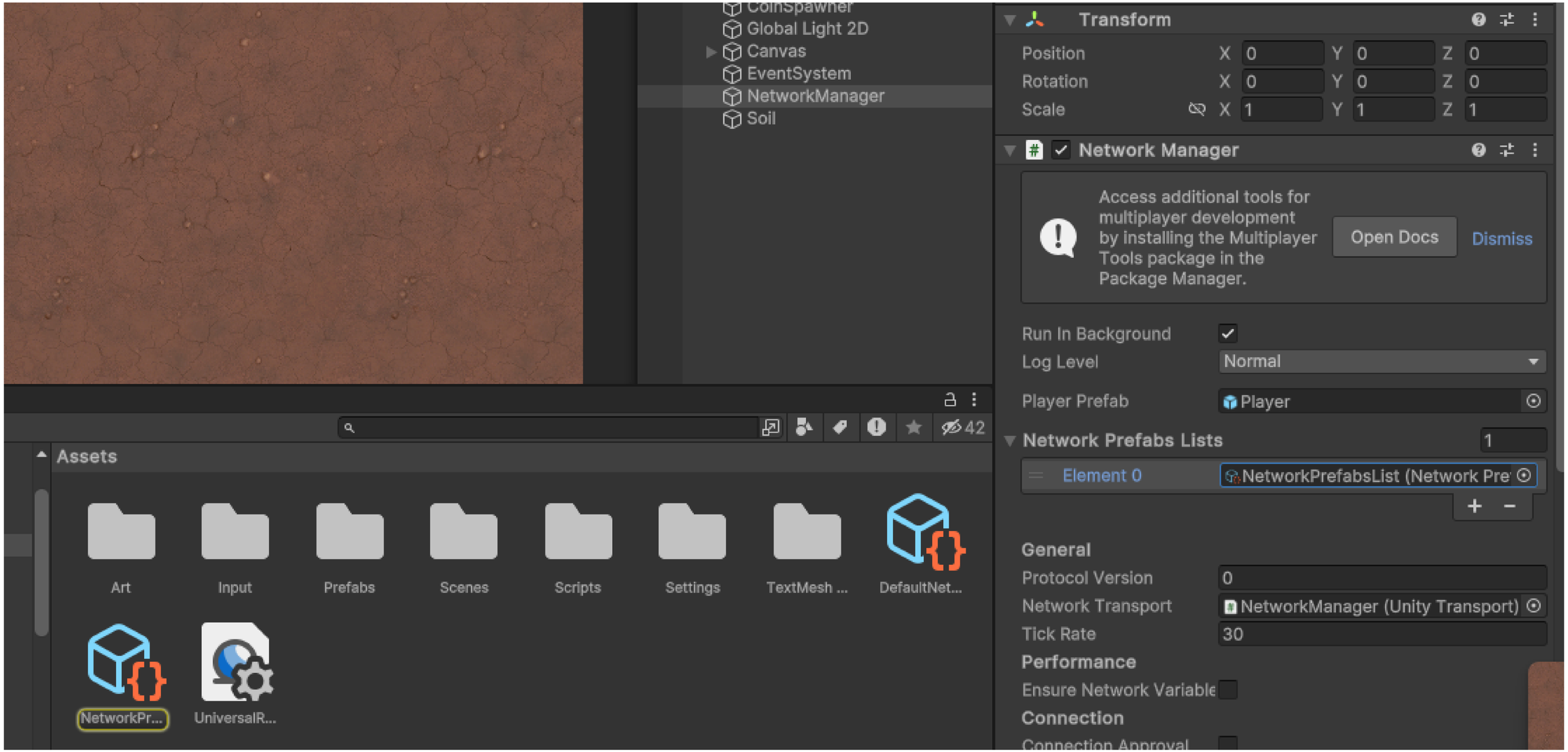
Network Prefabs List

2D

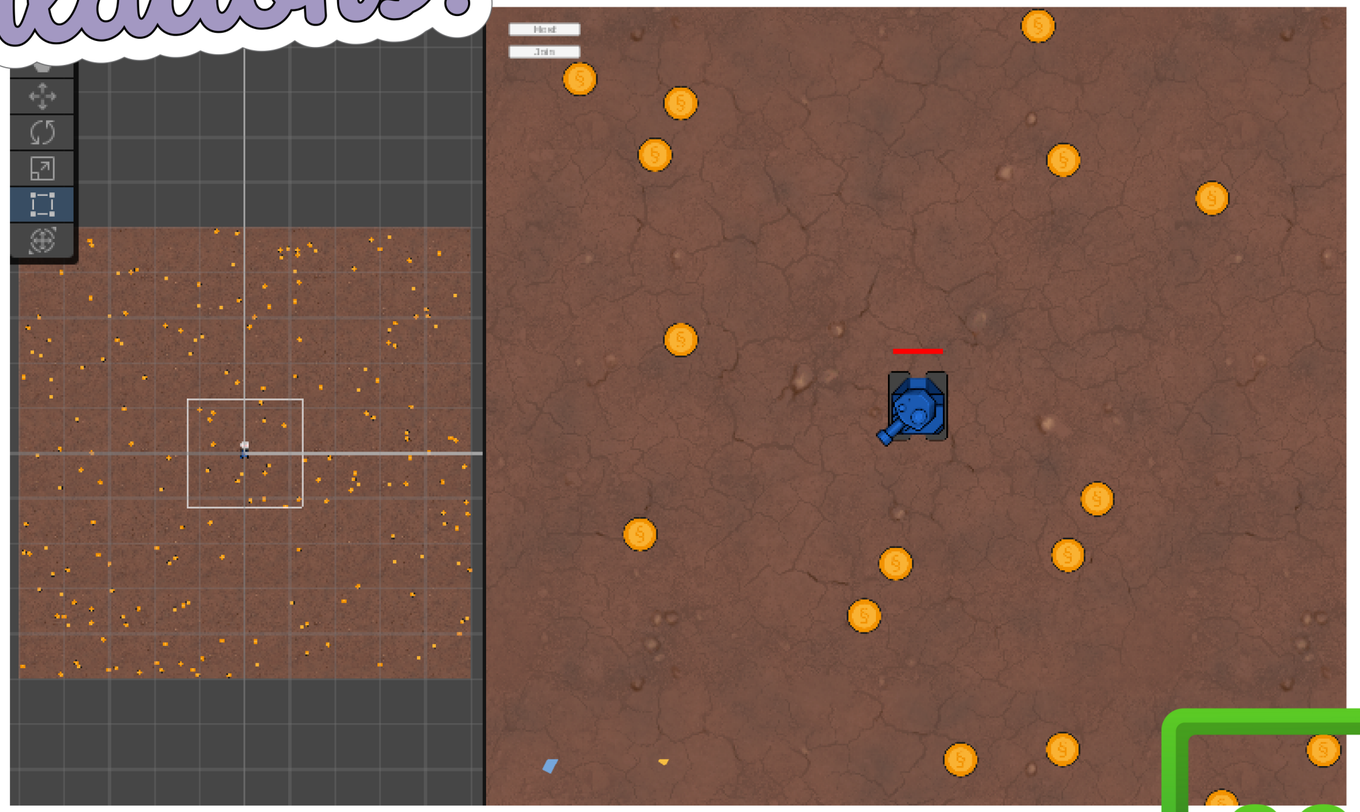
Create
Show in Explorer
Open
Delete
Rename
Copy Path Alt+Ctrl+C
Open Scene Additive
View in Package Manager
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All

Input
Netcode
Folder
C# Script
2D
Visual Scripting
Shader Graph
Shader
Shader Variant Collection
Testing
Playables
Assembly Definition
Assembly Definition Reference
Text
TextMeshPro
Scene
Scene Template
Scene Template From Scene
Volume Profile
Scene Template Pipeline
Prefab
Prefab Variant

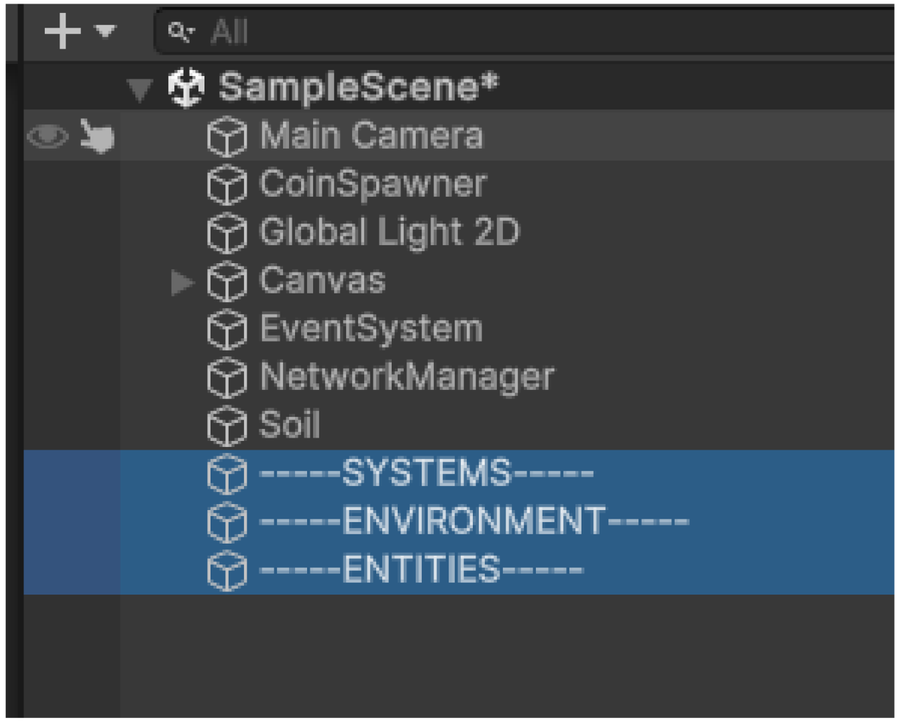

Inspector
Network Prefabs List (Network Prefabs List)
Open
Is Default
NetworkPrefabs
RespawningCoin
Override

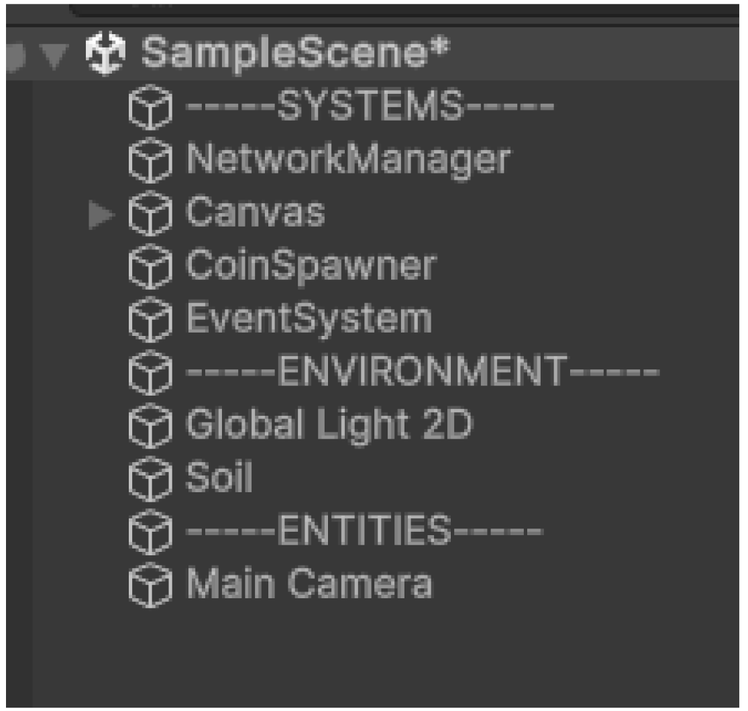

CoinSpawner
Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Network Manager
Access additional tools for multiplayer development by installing the Multiplayer Tools package in the Package Manager.
Open Docs
Dismiss
Run In Background
Log Level Normal
Player Prefab Player
Network Prefabs Lists 1
Element 0 NetworkPrefabsList (Network Pre
General
Protocol Version 0
Network Transport NetworkManager (Unity Transport)
Tick Rate 30
Performance
Ensure Network Variable
Connection
Connection Approval
42
Assets
Art
Input
Prefabs
Scenes
Scripts
Settings
TextMesh ...
DefaultNet...
NetworkPr...
UniversalR...

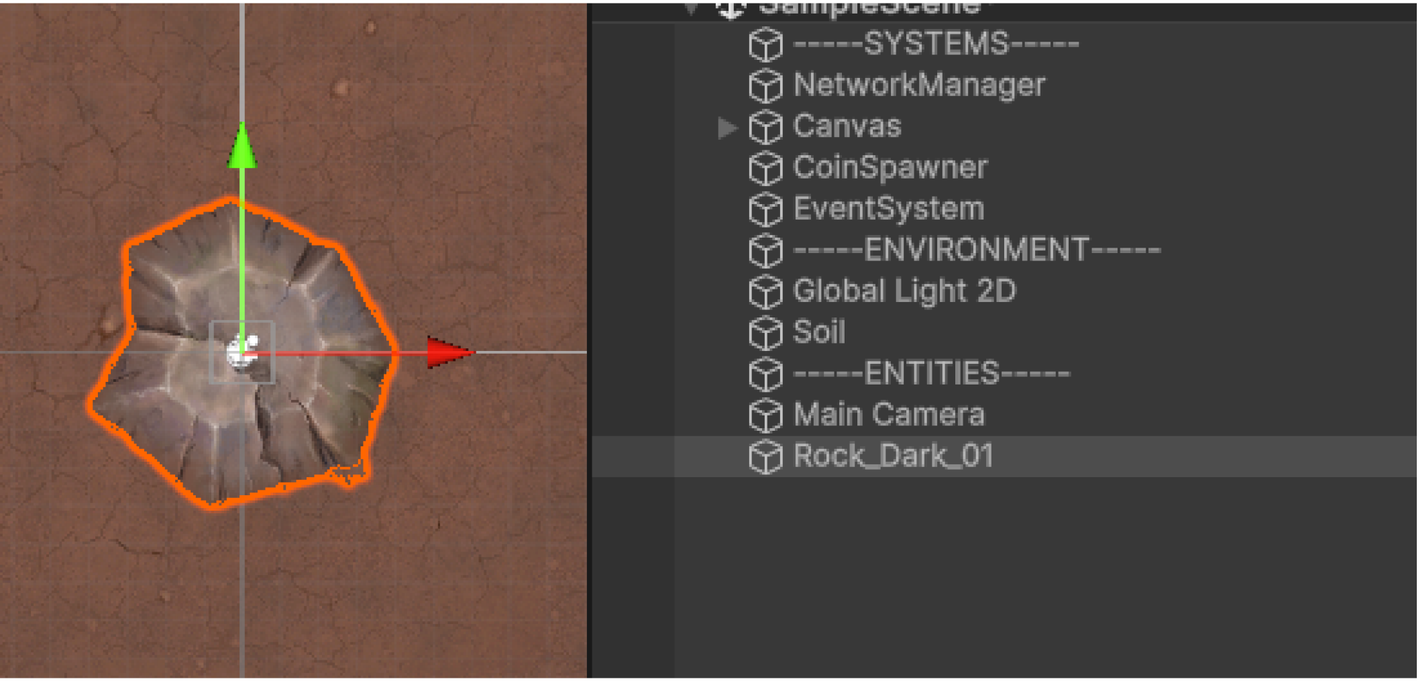
Coin Spawner

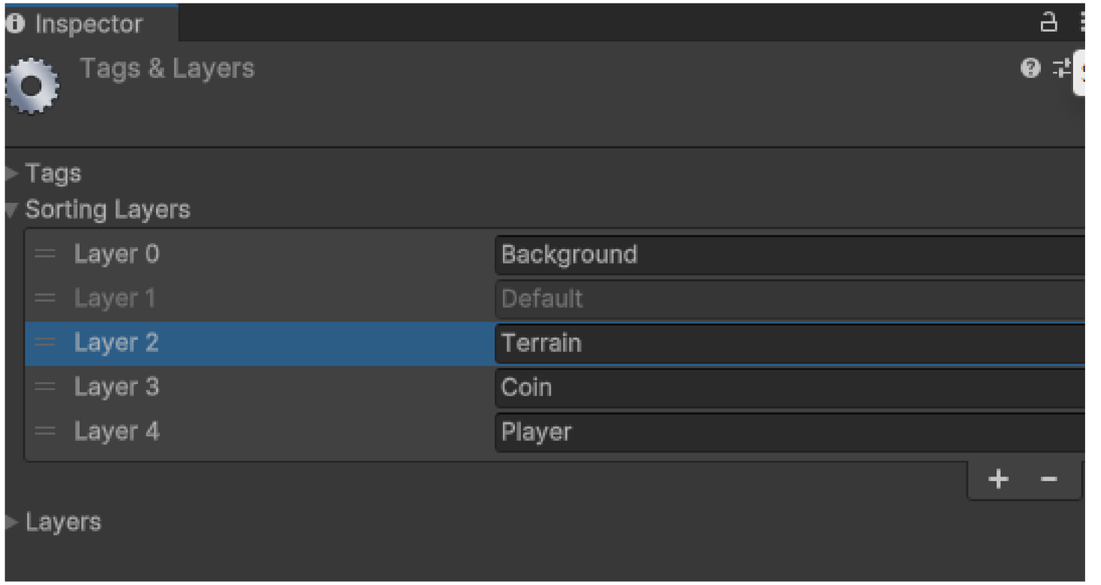


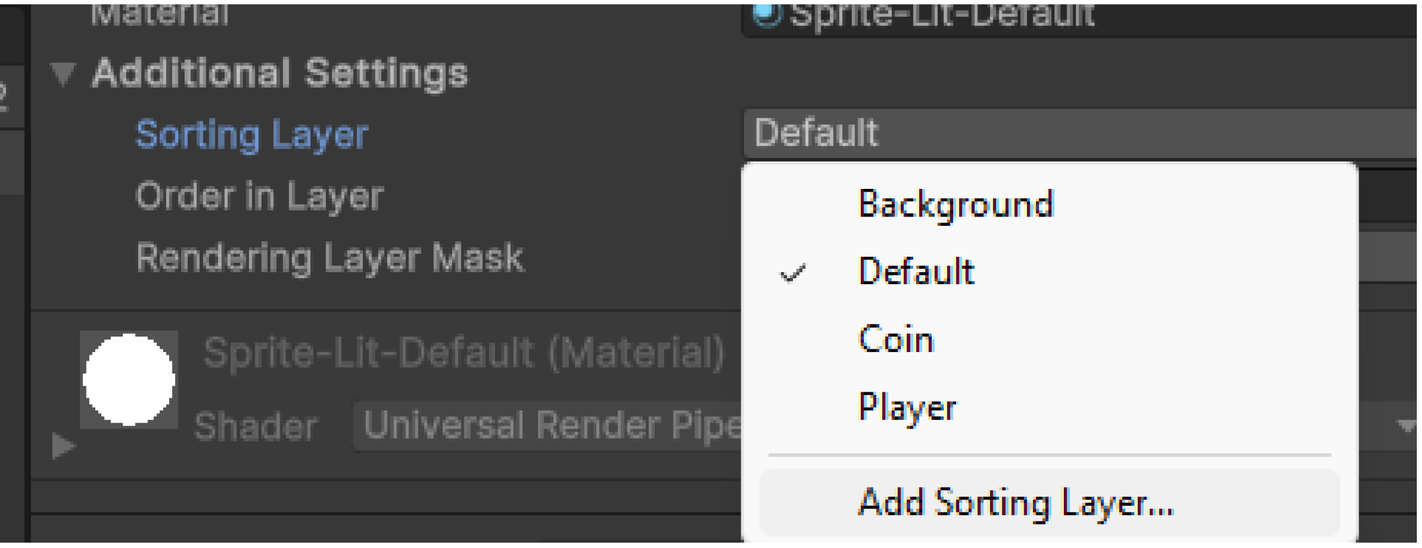
Map Design

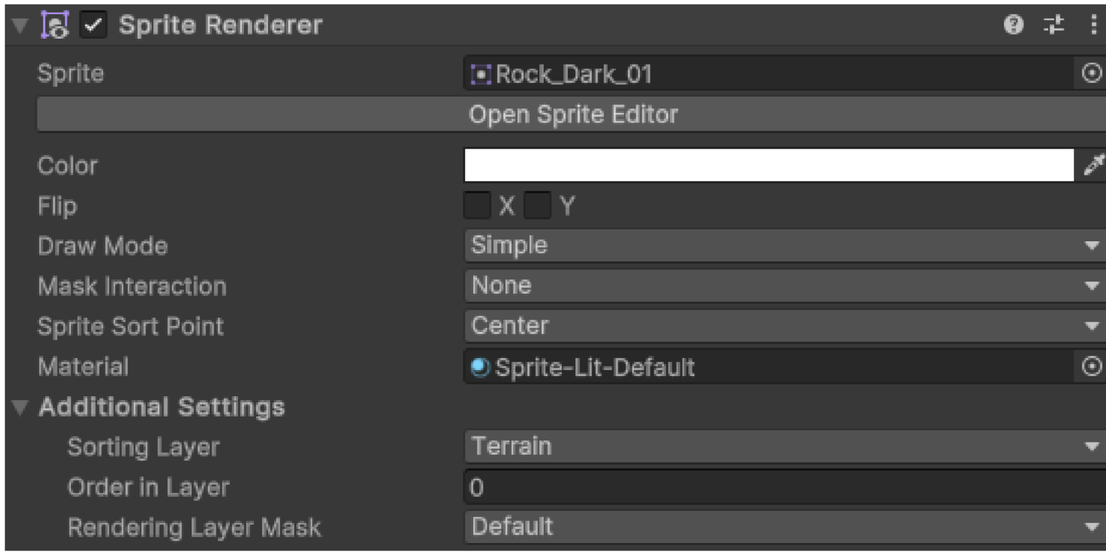

All
SampleScene*
Main Camera
CoinSpawner
Global Light 2D
Canvas
EventSystem
NetworkManager
Soil
-----SYSTEMS-----
-----ENVIRONMENT-----
-----ENTITIES-----

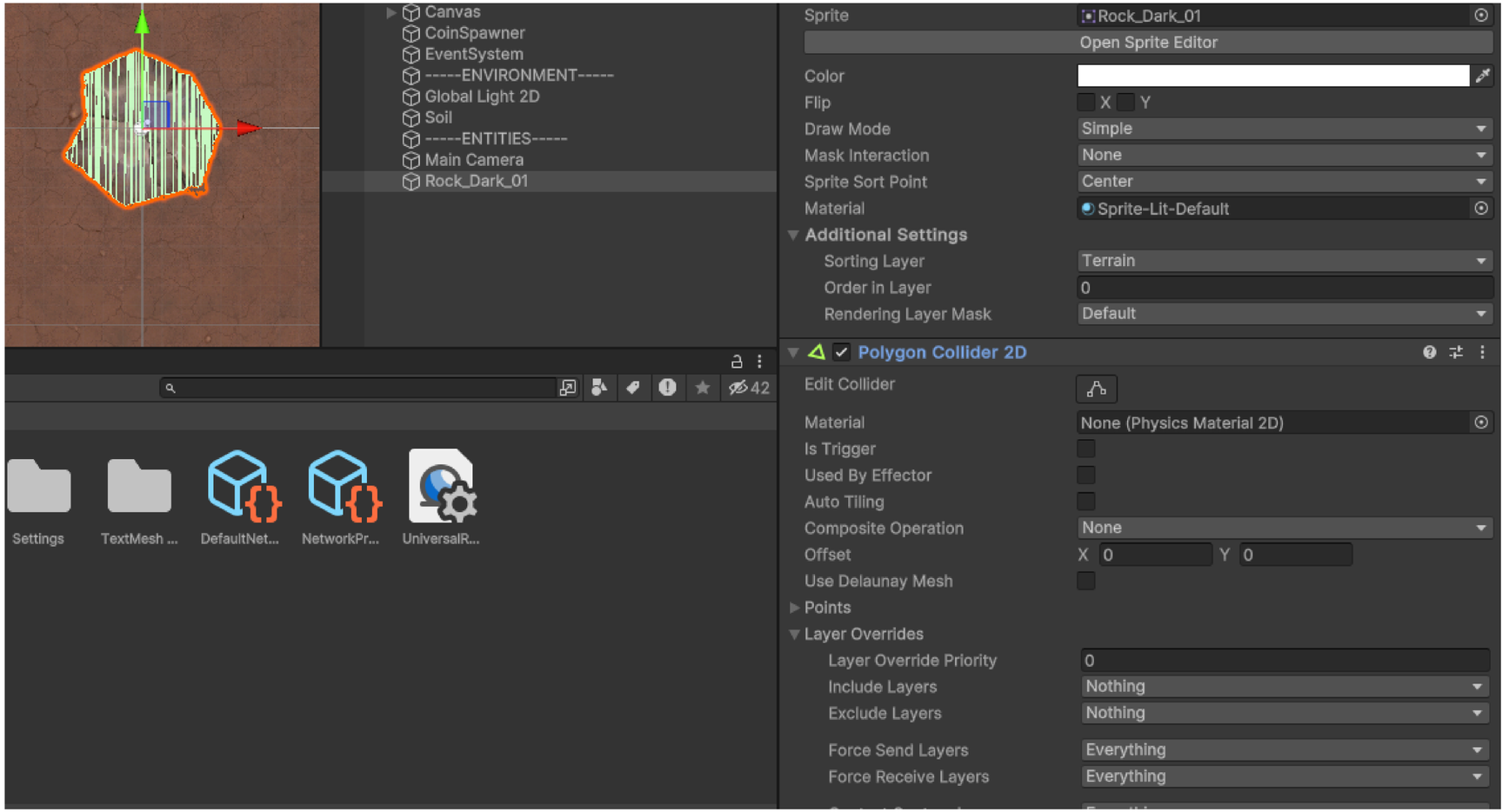

SampleScene*
-----SYSTEMS-----
NetworkManager
Canvas
CoinSpawner
EventSystem
-----ENVIRONMENT-----
Global Light 2D
Soil
-----ENTITIES-----
Main Camera

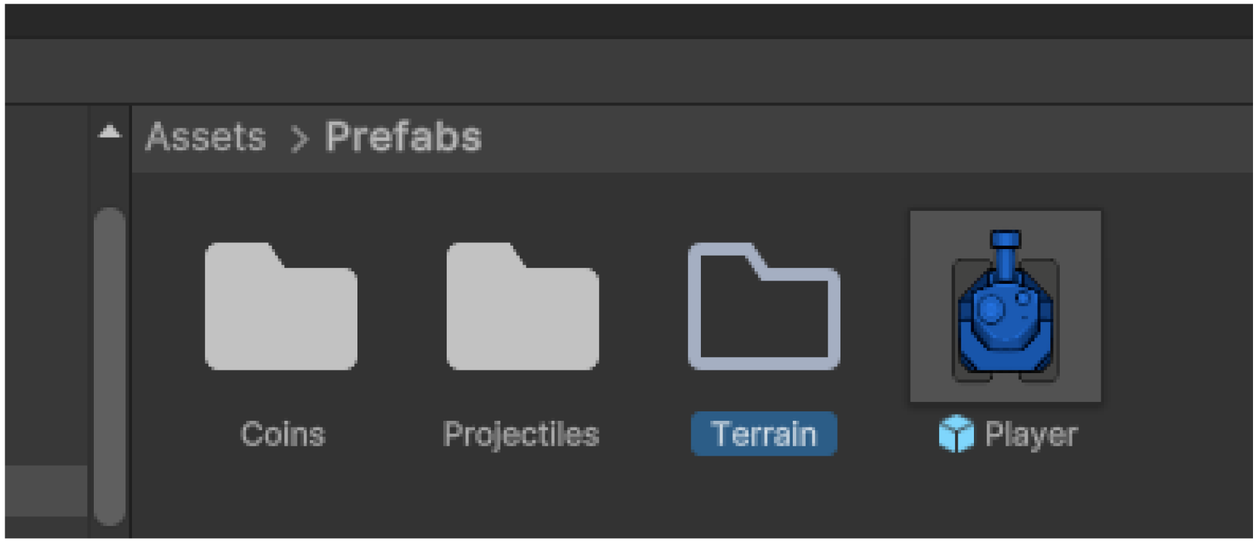

-----SYSTEMS-----
NetworkManager
Canvas
CoinSpawner
EventSystem
-----ENVIRONMENT-----
Global Light 2D
Soil
-----ENTITIES-----
Main Camera
Rock_Dark_01

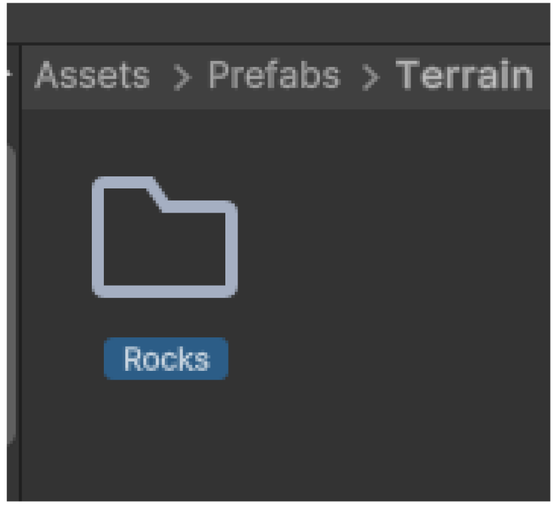

Inspector
Tags & Layers
Tags
Sorting Layers
Layer 0
Background
Layer 1
Default
Layer 2
Terrain
Layer 3
Coin
Layer 4
Player
Layers

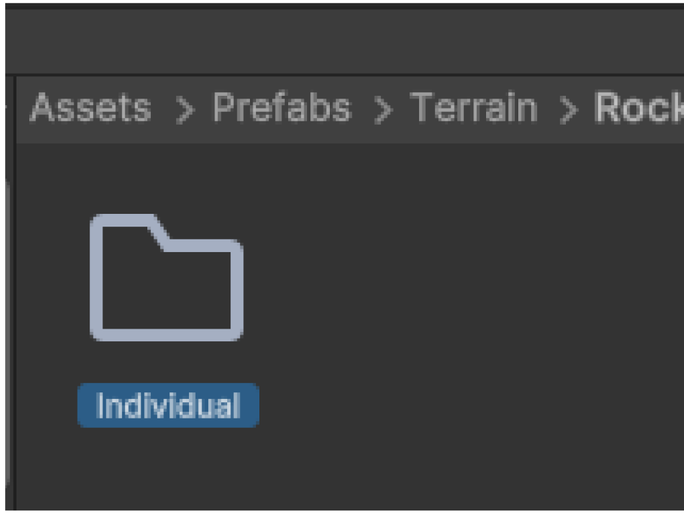

Material
Sprite-Lit-Default
Additional Settings
Sorting Layer
Default
Order in Layer
Rendering Layer Mask
Sprite-Lit-Default (Material)
Shader
Universal Render Pipe
Background
Default
Coin
Player
Add Sorting Layer...

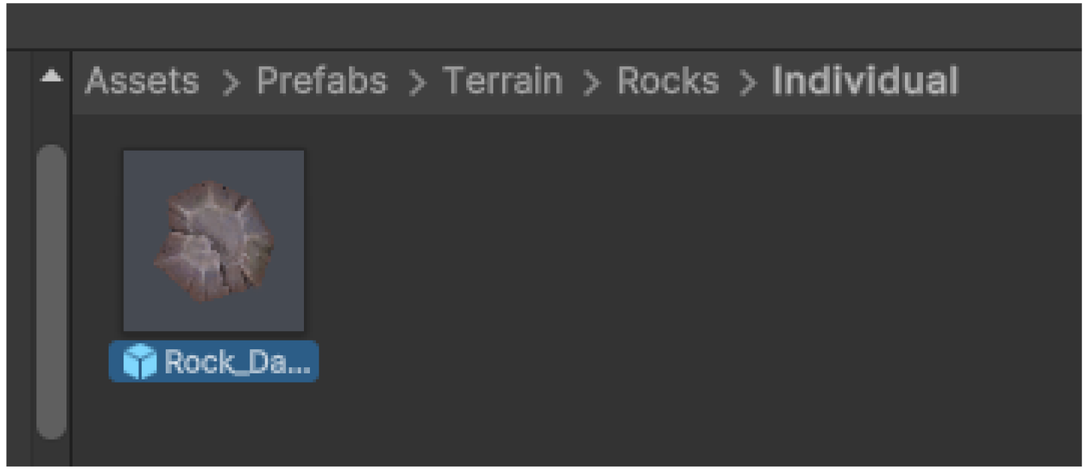

Sprite Renderer
Sprite
Rock_Dark_01
Open Sprite Editor
Color
Flip
X
Y
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Lit-Default
Additional Settings
Sorting Layer
Terrain
Order in Layer
0
Rendering Layer Mask
Default

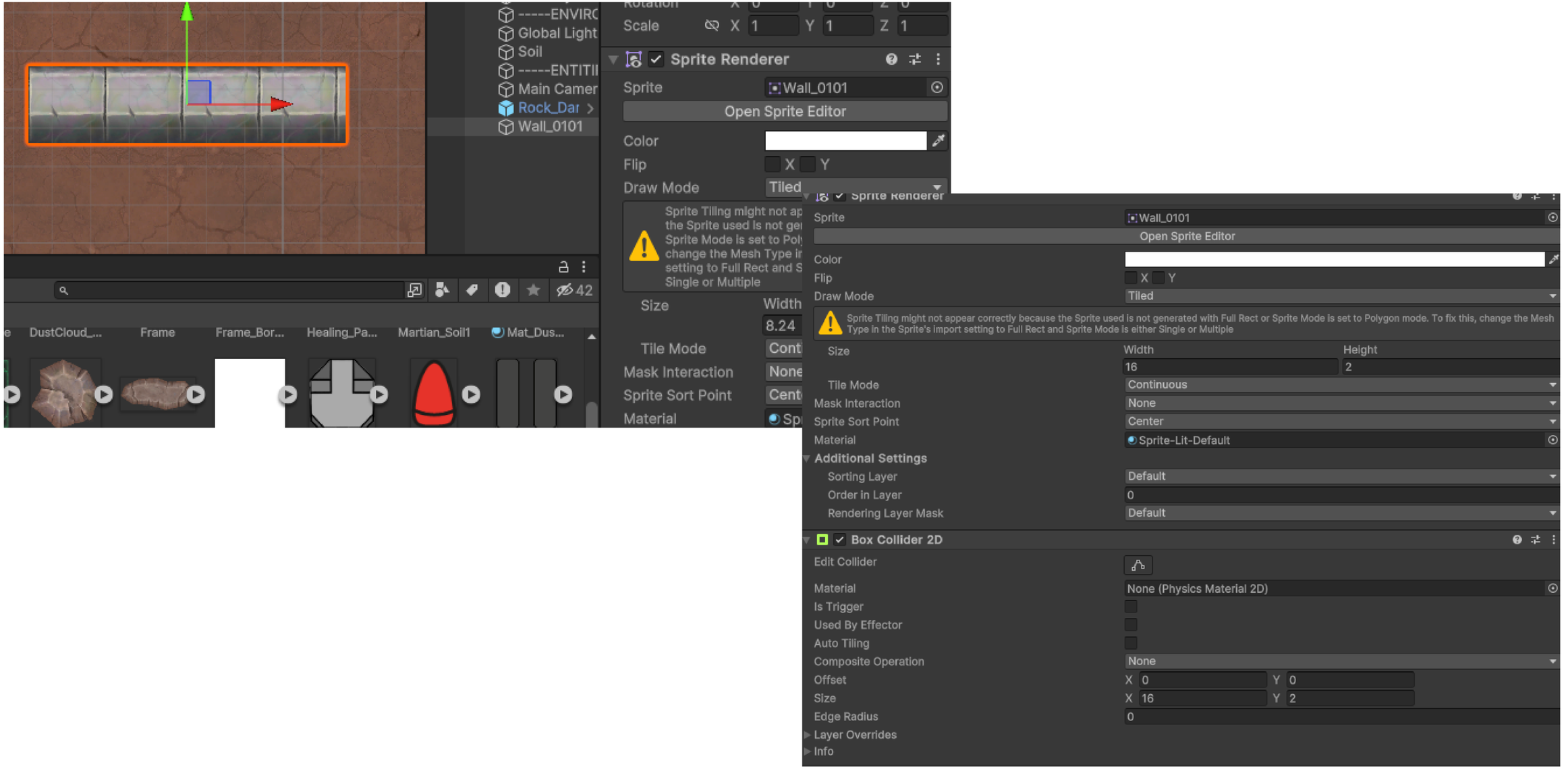

Canvas
CoinSpawner
EventSystem
-----ENVIRONMENT-----
Global Light 2D
Soil
-----ENTITIES-----
Main Camera
Rock_Dark_01
Settings
TextMesh ...
DefaultNet...
NetworkPr...
UniversalR...
Sprite
Rock_Dark_01
Open Sprite Editor
Color
Flip
X
Y
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Lit-Default
Additional Settings
Sorting Layer
Terrain
Order in Layer
0
Rendering Layer Mask
Default
Polygon Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Auto Tiling
Composite Operation
None
Offset
X 0
Y 0
Use Delaunay Mesh
Points
Layer Overrides
Layer Override Priority
0
Include Layers
Nothing
Exclude Layers
Nothing
Force Send Layers
Everything
Force Receive Layers
Everything


Assets > Prefabs
Coins
Projectiles
Terrain
Player


Assets > Prefabs > Terrain
Rocks


Assets > Prefabs > Terrain > Rock
Individual


Assets > Prefabs > Terrain > Rocks > Individual
Rock_Da...


Scale X 1 Y 1 Z 1
Sprite Renderer
Sprite Wall_0101
Open Sprite Editor
Color
Flip X Y
Draw Mode Tiled
Sprite Tiling might not appear correctly because the Sprite used is not generated with Full Rect or Sprite Mode is set to Polygon mode. To fix this, change the Mesh Type in the Sprite's import setting to Full Rect and Sprite Mode is either Single or Multiple
Size Width Height
16 2
Tile Mode Continuous
Mask Interaction None
Sprite Sort Point Center
Material Sprite-Lit-Default
Additional Settings
Sorting Layer Default
Order in Layer 0
Rendering Layer Mask Default
Box Collider 2D
Edit Collider
Material None (Physics Material 2D)
Is Trigger
Used By Effector
Auto Tiling
Composite Operation None
Offset X 0 Y 0
Size X 16 Y 2
Edge Radius 0
Layer Overrides
Info
Global Light
Soil
Main Camer
Rock_Dar
Wall_0101
DustCloud_...
Frame
Frame_Bor...
Healing_Pa...
Martian_Soil1
Mat_Dus...

Create Your Own Map

- Use assets to create your own map
- Find other assets to create a map
- Make your own assets to create a map

Map Design

Assignment

ให้ทำการถ่ายคลิปผลลัพธ์ของการทำ Workshop ตามเนื้อหา Workshop ในแต่ละหัวข้อนี้พร้อมอธิบายประกอบตามความเข้าใจ

- Health Component
- Health Display
- Dealing Damage
- Coins
- Coin Wallet
- Coin Spawner
- Map Design

Next Week

Update Progress 1st Presentation

https://discord.gg/24xmUFHfzR

WOLVEDEN ACADEMY

THANK YOU

facebook.com/GameDevMew