

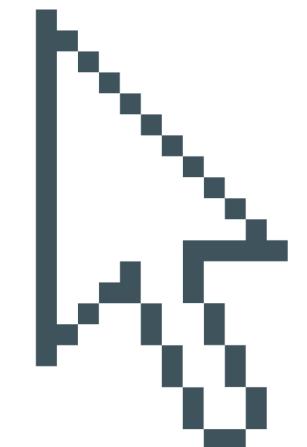


WOLVEDEN ACADEMY

NETWORKING AND MULTIPLAYER ONLINE GAMES



MULTIPLAYER NEW ERA



BY PONGSATHORN KIATTICHAOENPORN (MEW)

WEEK 11 : GAMEPLAY ADDITIONS 3

Gameplay Additions 3

Topic in this week

Workshop

- Leaderboard Names
- Leaderboard Updating
- Leaderboard Sorting
- Losing Coins
- Bounty Coins

Leaderboard Names



Leaderboard

Leaderboar...

Leaderboar...

Assets/Scripts/UI/Leaderboard/LeaderboardEntityDisplay.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using TMPro;
4  using UnityEngine;
5
6  public class LeaderboardEntityDisplay : MonoBehaviour
7  {
8    [SerializeField] private TMP_Text displayText;
9  }
```

[T] TextMeshPro - Text (UI)

Text Input

1. Name (20)

LeaderboardEntity

Canvas (Environment)

LeaderboardEntity

Auto Size Options Min 18 Max 36 WD% 0 Line 0

Vertex Color

Color Gradient

Override Tags

Spacing Options (em) Character 0 Word 0
Line 0 Paragraph 0

Alignment

Wrapping

Overflow

Horizontal Mapping

Vertical Mapping

Extra Settings *(Click to expand)*

Leaderboard Entity Display (Script) ⓘ ⚡ :

Script LeaderboardEntityDisplay

Display Text LeaderboardEntity (Text Mesh Pro)

Leaderboard Entity Display

- Create a **public void Initialise** method that takes in the client's id, name, and coins.
- Store the values in private variables
- Update the text to be in the format: “**1. Name (Coins)**” or whichever format you prefer

C# LeaderboardEntityDisplay.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using TMPro;
4  using Unity.Collections;
5  using UnityEngine;
6
7  public class LeaderboardEntityDisplay : MonoBehaviour
8  {
9      [SerializeField] private TMP_Text displayText;
10
11     private FixedString32Bytes playerName;
12
13     public ulong ClientId { get; private set; }
14     public int Coins { get; private set; }
15
16     public void Initialise(ulong clientId, FixedString32Bytes playerName, int coins)
17     {
18         ClientId = clientId;
19         this.playerName = playerName;
20         UpdateCoins(coins);
21     }
22
23     public void UpdateCoins(int coins)
24     {
25         Coins = coins;
26         UpdateText();
27     }
28
29     private void UpdateText()
30     {
31         displayText.text = $"1. {playerName} ({Coins})";
32     }
33 }
```

C# Leaderboard.cs

```
⊕ Unity Script (1 asset reference) | 0 references
public class Leaderboard : NetworkBehaviour
{
    [SerializeField] private Transform leaderboardEntityHolder;
    [SerializeField] private LeaderboardEntityDisplay leaderboardEntityPrefab;

    private NetworkList<LeaderboardEntityState> leaderboardEntities;
    private List<LeaderboardEntityDisplay> entityDisplays = new List<LeaderboardEntityDisplay>();
```

3 references

```
private void HandleLeaderboardEntitiesChanged(NetworkListEvent<LeaderboardEntityState> changeEvent)
```

```
{
```

```
    switch (changeEvent.Type)
```

```
{
```

```
    case NetworkListEvent<LeaderboardEntityState>.EventType.Add:
```

```
        if(!entityDisplays.Any(x=>x.ClientId == changeEvent.Value.ClientId))
```

```
{
```

```
            LeaderboardEntityDisplay leaderboardEntity =
                Instantiate(leaderboardEntityPrefab, leaderboardEntityHolder);
            leaderboardEntity.Initialise(
                changeEvent.Value.ClientId,
                changeEvent.Value.PlayerName,
                changeEvent.Value.Coins);
            entityDisplays.Add(leaderboardEntity);
        }
```

```
        break;
```

```
    case NetworkListEvent<LeaderboardEntityState>.EventType.Remove:
```

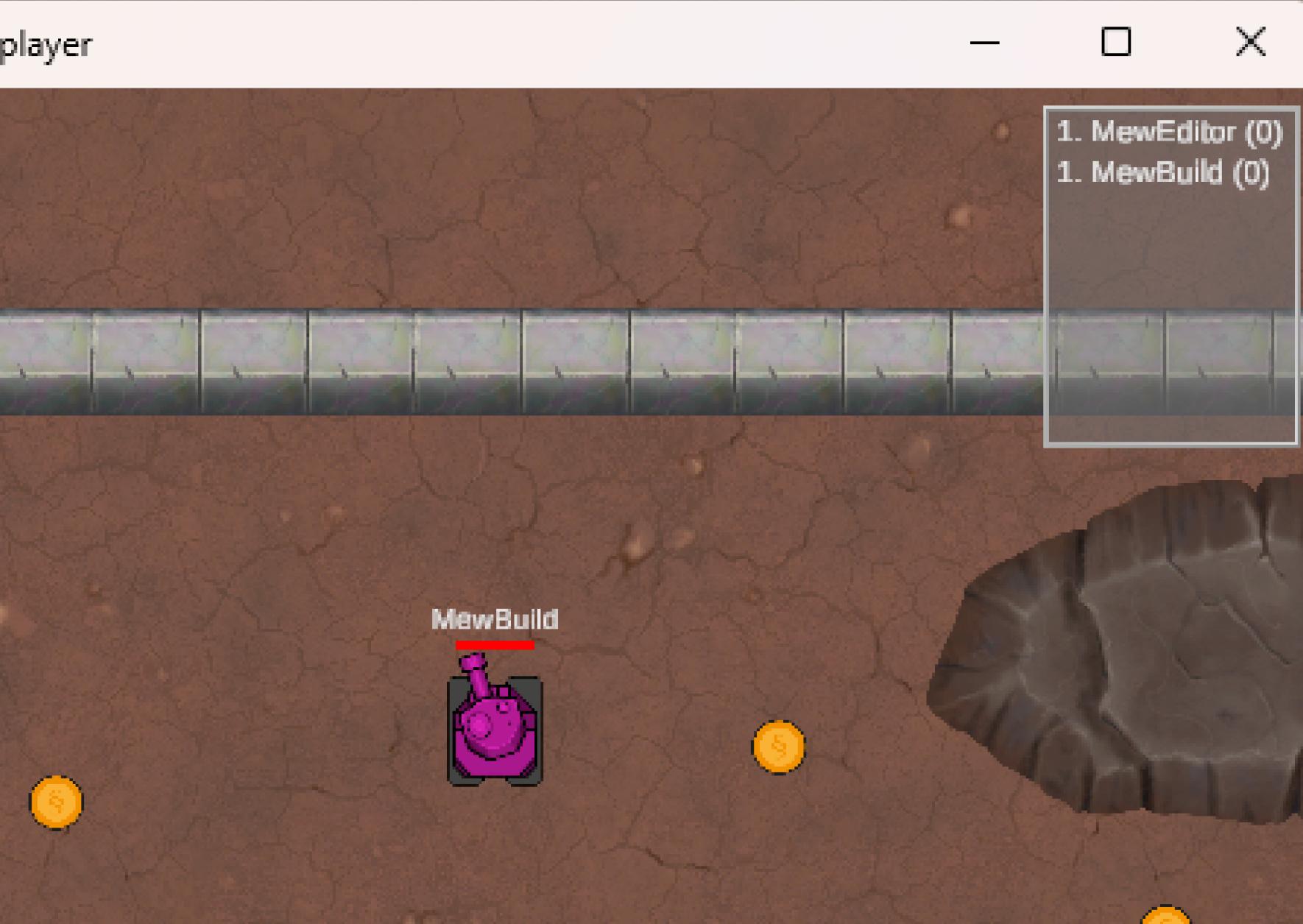
```
        LeaderboardEntityDisplay displayToRemove =
            entityDisplays.FirstOrDefault(x => x.ClientId == changeEvent.Value.ClientId);
```

```
        if (displayToRemove != null)
```

```
        {
            displayToRemove.transform.SetParent(null);
            Destroy(displayToRemove.gameObject);
            entityDisplays.Remove(displayToRemove);
        }
```

```
        break;
```

```
private void HandleLeaderboardEntitiesChanged(NetworkListEvent<LeaderboardEntityState> changeEvent)
{
    switch (changeEvent.Type)
    {
        case NetworkListEvent<LeaderboardEntityState>.EventType.Add:
            if(!entityDisplays.Any(x=>x.ClientId == changeEvent.Value.ClientId))
            {
                LeaderboardEntityDisplay leaderboardEntity =
                    Instantiate(leaderboardEntityPrefab, leaderboardEntityHolder);
                leaderboardEntity.Initialise(
                    changeEvent.Value.ClientId,
                    changeEvent.Value.PlayerName,
                    changeEvent.Value.Coins);
                entityDisplays.Add(leaderboardEntity);
            }
            break;
        case NetworkListEvent<LeaderboardEntityState>.EventType.Remove:
            LeaderboardEntityDisplay displayToRemove =
                entityDisplays.FirstOrDefault(x => x.ClientId == changeEvent.Value.ClientId);
            if (displayToRemove != null)
            {
                displayToRemove.transform.SetParent(null);
                Destroy(displayToRemove.gameObject);
                entityDisplays.Remove(displayToRemove);
            }
            break;
        case NetworkListEvent<LeaderboardEntityState>.EventType.Value:
            LeaderboardEntityDisplay displayToUpdate =
                entityDisplays.FirstOrDefault(x => x.ClientId == changeEvent.Value.ClientId);
            if (displayToUpdate != null)
            {
                displayToUpdate.UpdateCoins(changeEvent.Value.Coins);
            }
            break;
    }
}
```



Leaderboard Updating

C# TankPlayer.cs

```
⊕ Unity Script (1 asset reference) | 23 references
[-]public class TankPlayer : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private CinemachineVirtualCamera virtualCamera;

    [SerializeField] private Texture2D crosshair;
    2 references
    [field:SerializeField] public Health Health { get; private set; }
    0 references
    [field:SerializeField] public CoinWallet Wallet { get; private set; }

    [Header("Variables")]
    [SerializeField] private float health;
    1 reference
    [SerializeField] private float maxHealth;
    1 reference
}
```

The screenshot shows the Unity Editor's Player Inspector window for a GameObject named "Player".

Player Components:

- TankTreads
- TankBody
- DustCloud
- RightTracks
- LeftTracks
- TurretPivot
- Turret
 - ProjectileSpawnPoint
 - MuzzleFlash
- OverheadCanvas
 - HealthBarBG
 - HealthBar
 - DisplayNameText
- FollowCam

Tank Player (Script) Settings:

- Type not renderable
- Player Color Index: 0
- Script: TankPlayer
- References:**
 - Virtual Camera: FollowCam (Cinemachine Virtual Cam)
 - Crosshair: Crosshair_White
 - Health: Player (Health)
 - Wallet: Player (Coin Wallet)
- Settings:**
 - Owner Priority: 15

C# Leaderboard.cs

```
0 references
121     |
122     |
123     |
124 }
```

```
private void HandlePlayerSpawned(TankPlayer player)
{
    leaderboardEntities.Add(new LeaderboardEntityState
    {
        ClientId = player.OwnerClientId,
        PlayerName = player.PlayerName.Value,
        Coins = 0
    });

    player.Wallet.TotalCoins.OnValueChanged += (oldCoins, newCoins) =>
        HandleCoinsChanged(player.OwnerClientId, newCoins);
}
```

```
2 references
private void HandlePlayerDespawned(TankPlayer player)
{
    foreach (LeaderboardEntityState entity in leaderboardEntities)
    {
        if(entity.ClientId != player.OwnerClientId) { continue; }
        leaderboardEntities.Remove(entity);
        break;
    }

    player.Wallet.TotalCoins.OnValueChanged -= (oldCoins, newCoins) =>
        HandleCoinsChanged(player.OwnerClientId, newCoins);
}
```

C# Leaderboard.cs

```
private void HandleCoinsChanged(ulong clientId, int newCoins)
{
    for (int i = 0; i < leaderboardEntities.Count; i++)
    {
        if (leaderboardEntities[i].ClientId != clientId) { continue; }

        leaderboardEntities[i] = new LeaderboardEntityState
        {
            ClientId = leaderboardEntities[i].ClientId,
            PlayerName = leaderboardEntities[i].PlayerName,
            Coins = newCoins
        };

        return;
    }
}
```



Leaderboard Sorting

C# Leaderboard.cs

```
⊕ Unity Script (1 asset reference) | 0 references
8  public class Leaderboard : NetworkBehaviour
9  {
10     [SerializeField] private Transform leaderboardEntityHolder;
11     [SerializeField] private LeaderboardEntityDisplay leaderboardEntityPrefab;
12     [SerializeField] private int entitiesToDisplay = 8;
13
14     private NetworkList<LeaderboardEntityState> leaderboardEntities;
15     private List<LeaderboardEntityDisplay> entityDisplays = new List<LeaderboardEntityDisplay>();
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61     private void HandleLeaderboardEntitiesChanged(NetworkListEvent<LeaderboardEntityState> changeEvent)
62     {
63         switch (changeEvent.Type)
64         {
65             case NetworkListEvent<LeaderboardEntityState>.EventType.Add: ...
66             case NetworkListEvent<LeaderboardEntityState>.EventType.Remove: ...
67             case NetworkListEvent<LeaderboardEntityState>.EventType.Value: ...
68         }
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97     entityDisplays.Sort((x, y) => y.Coins.CompareTo(x.Coins));
98
99     for(int i=0;i<entityDisplays.Count; i++)
100     {
101         entityDisplays[i].transform.SetSiblingIndex(i);
102         entityDisplays[i].UpdateText();
103     }
104 }
```

C# LeaderboardEntityDisplay.cs

```
public void UpdateText()
{
    displayText.text = $"{transform.GetSiblingIndex() + 1}. {playerName} ({coins})";
}
```

C# Leaderboard.cs

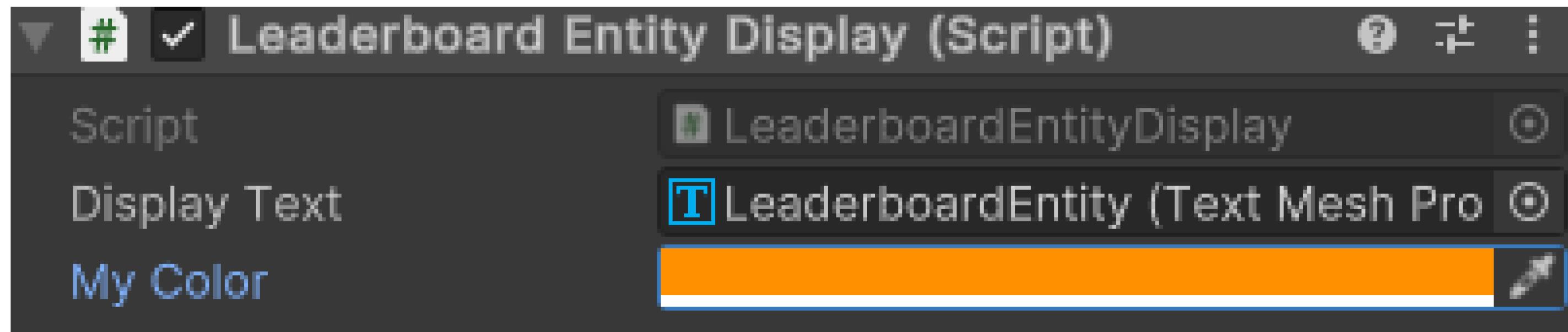
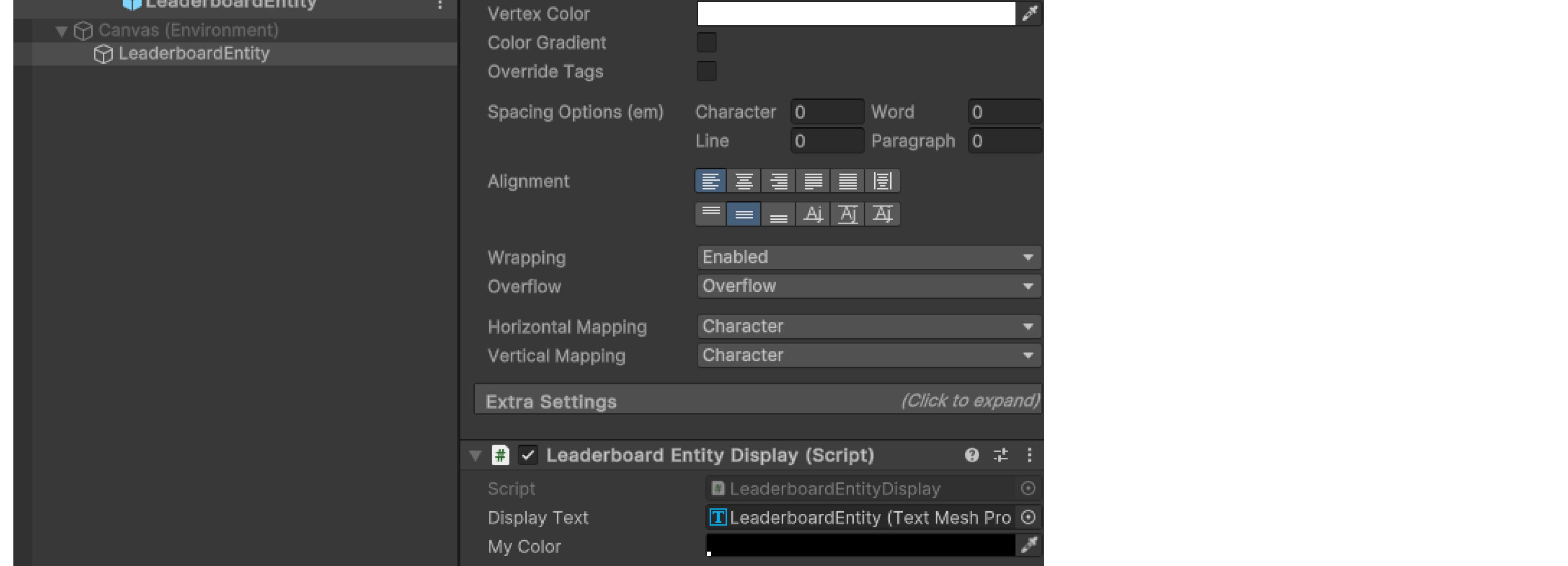
```
61     private void HandleLeaderboardEntitiesChanged(NetworkListEvent<LeaderboardEntityState> changeEvent)
62     {
63         switch (changeEvent.Type)
64         {
65             case NetworkListEvent<LeaderboardEntityState>.EventType.Add: ...
66             case NetworkListEvent<LeaderboardEntityState>.EventType.Remove: ...
67             case NetworkListEvent<LeaderboardEntityState>.EventType.Value: ...
68         }
69
70
71         entityDisplays.Sort((x, y) => y.Coins.CompareTo(x.Coins));
72
73
74         for(int i=0;i<entityDisplays.Count; i++)
75         {
76             entityDisplays[i].transform.SetSiblingIndex(i);
77             entityDisplays[i].UpdateText();
78             entityDisplays[i].gameObject.SetActive(i <= entitiesToDisplay - 1);
79
80
81         }
82
83
84         LeaderboardEntityDisplay myDisplay =
85             entityDisplays.FirstOrDefault(x => x.ClientId == NetworkManager.Singleton.LocalClientId);
86
87
88         if(myDisplay != null)
89         {
90             if(myDisplay.transform.GetSiblingIndex() >= entitiesToDisplay)
91             {
92                 leaderboardEntityHolder.GetChild(entitiesToDisplay-1).gameObject.SetActive(false);
93                 myDisplay.gameObject.SetActive(true);
94             }
95         }
96
97     }
98
99 }
```

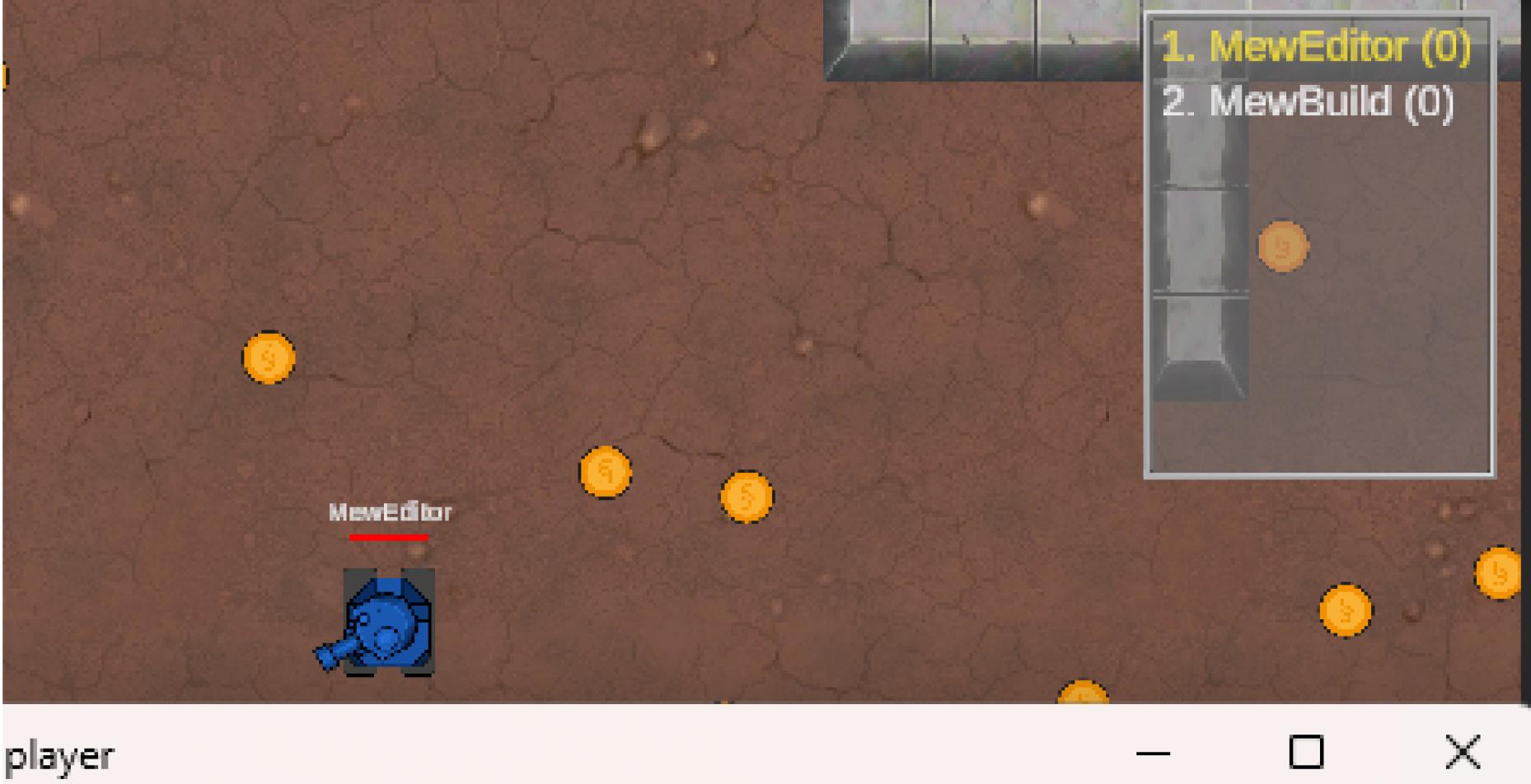
C# LeaderboardEntityDisplay.cs

```
④ Unity Script (1 asset reference) / References
[-]public class LeaderboardEntityDisplay : MonoBehaviour
  {
    [SerializeField] private TMP_Text displayText;
    [SerializeField] private Color myColor;
```

C# LeaderboardEntityDisplay.cs

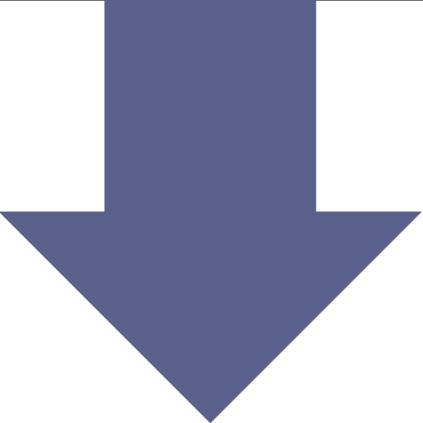
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using TMPro;
4  using Unity.Collections;
5  using Unity.Netcode;
6  using UnityEngine;
7
8  public class LeaderboardEntityDisplay : MonoBehaviour
9  {
10     [SerializeField] private TMP_Text displayText;
11     [SerializeField] private Color myColor;
12
13     private FixedString32Bytes playerName;
14
15     public ulong ClientId { get; private set; }
16     public int Coins { get; private set; }
17
18     public void Initialise(ulong clientId, FixedString32Bytes playerName, int coins)
19     {
20         ClientId = clientId;
21         this.playerName = playerName;
22
23         if(clientId == NetworkManager.Singleton.LocalClientId)
24         {
25             displayText.color = myColor;
26         }
27     }
28 }
```





Losing Coins

```
UnityScript (PlayerReference) References
public class RespawnHandler : NetworkBehaviour
{
    [SerializeField] private NetworkObject playerPrefab;
```

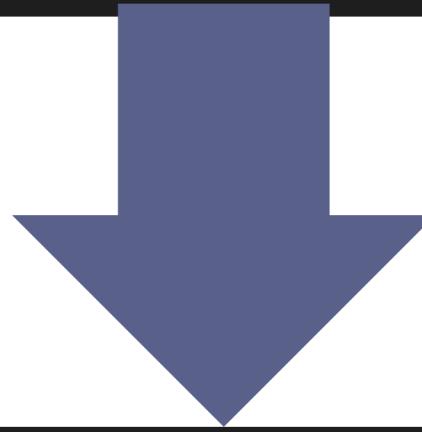


```
public class RespawnHandler : NetworkBehaviour
{
    [SerializeField] private TankPlayer playerPrefab;
```

```
private IEnumerator RespawnPlayer(ulong ownerId)
{
    yield return null;

    NetworkObject playerInstance = Instantiate(
        playerPrefab, SpawnPoint.GetRandomSpawnPos(), Quaternion.identity);

    playerInstance.SpawnAsPlayerObject(ownerId);
}
```



```
1 reference
private IEnumerator RespawnPlayer(ulong ownerId)
{
    yield return null;

    TankPlayer playerInstance = Instantiate(
        playerPrefab, SpawnPoint.GetRandomSpawnPos(), Quaternion.identity);

    playerInstance.NetworkObject.SpawnAsPlayerObject(ownerId);
}
```

Unity Script (1 asset reference) | 0 references

```
public class RespawnHandler : NetworkBehaviour
{
    [SerializeField] private TankPlayer playerPrefab;
    [SerializeField] private float keptCoinPercentage;
```

Losing Coins

- Divide the player's coins and store it in a variable
- Pass the calculated coin value through to the **RespawnPlayer** coroutine
- Add the calculated coin value to the new playerInstance's **CoinWallet**'s total coins value

```
private void HandlePlayerDie(TankPlayer player)
{
    int keptCoins = (int)(player.Wallet.TotalCoins.Value * (keptCoinPercentage / 100));

    Destroy(player.gameObject);

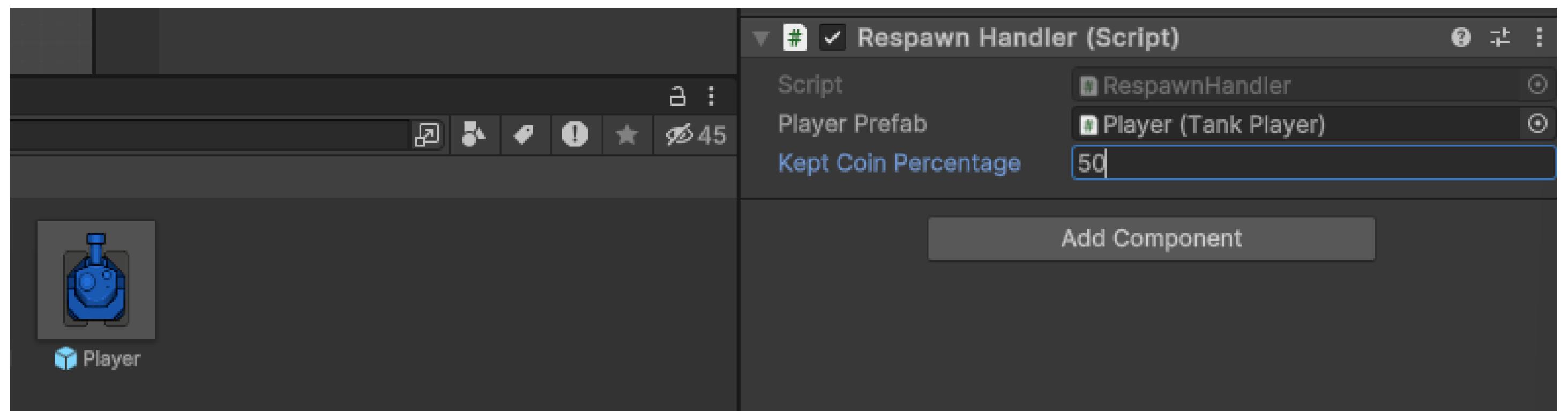
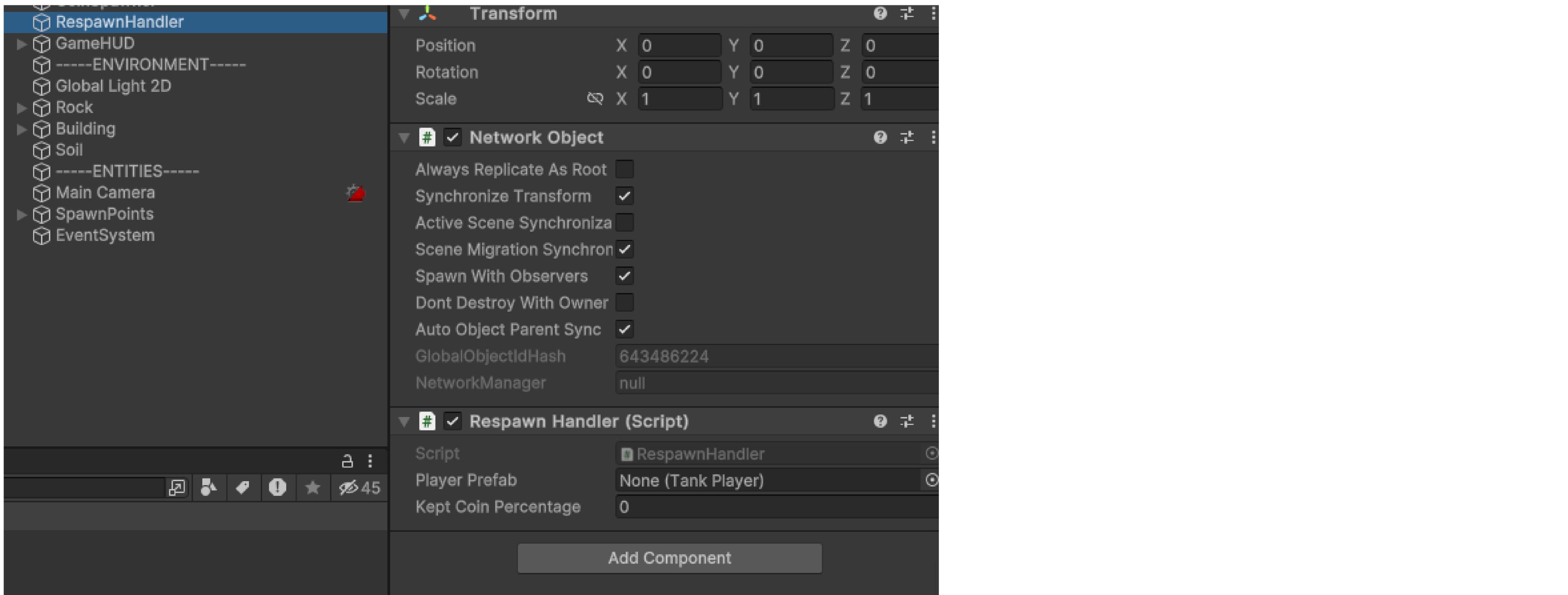
    StartCoroutine(RespawnPlayer(player.OwnerClientId, keptCoins));
}
```

1 reference

```
private IEnumerator RespawnPlayer(ulong ownerId, int keptCoins)
{
    yield return null;

    TankPlayer playerInstance = Instantiate(
        playerPrefab, SpawnPoint.GetRandomSpawnPos(), Quaternion.identity);
    playerInstance.Wallet.TotalCoins.Value += keptCoins;

    playerInstance.NetworkObject.SpawnAsPlayerObject(ownerId);
}
```



```
1 reference
private IEnumerator RespawnPlayer(ulong ownerId, int keptCoins)
{
    yield return null;

    TankPlayer playerInstance = Instantiate(
        playerPrefab, SpawnPoint.GetRandomSpawnPos(), Quaternion.identity);
    playerInstance.Wallet.TotalCoins.Value += keptCoins;

    playerInstance.NetworkObject.SpawnAsPlayerObject(ownerId);
}
```

 [10:13:45] NetworkVariable is written to, but doesn't know its NetworkBehaviour yet. Are you modifying a NetworkVariable before the NetworkObject is spawned?
UnityEngine.Debug:LogWarning (object)
UnityEngine.Debug:LogWarning (object)
Unity.Netcode.NetworkVariableBase:SetDirty (bool) (at

สาเหตุก็คือเราทำการ Modify ค่าก่อนที่จะสร้าง PlayerObject

```
1 reference
private IEnumerator RespawnPlayer(ulong ownerId, int keptCoins)
{
    yield return null;

    TankPlayer playerInstance = Instantiate(
        playerPrefab, SpawnPoint.GetRandomSpawnPos(), Quaternion.identity);

    playerInstance.NetworkObject.SpawnAsPlayerObject(ownerId);

    playerInstance.Wallet.TotalCoins.Value += keptCoins;
}
```

วิธีการแก้ก็คือ สั่ง Modify ค่า หลัง Spawn Object ไปแล้วนั่นเอง

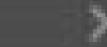


Bounty Coins



CoinBase

- SpawnPoints
- EventSystem
- CoinBase



a :



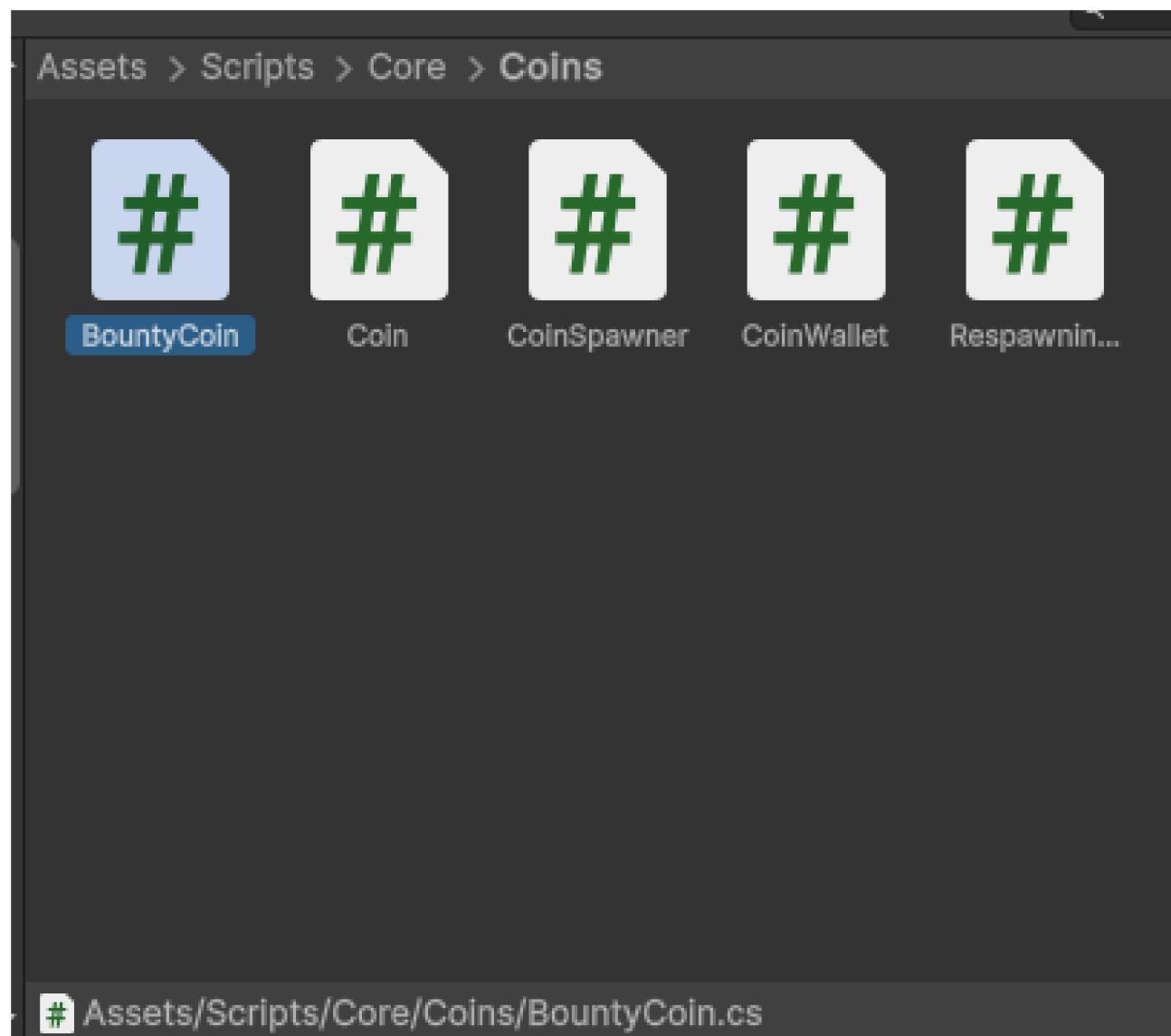
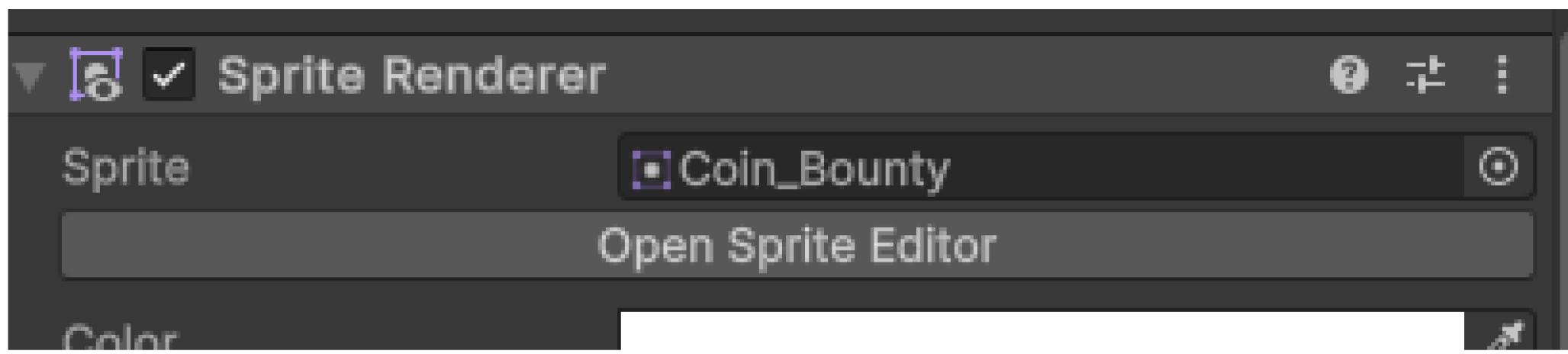
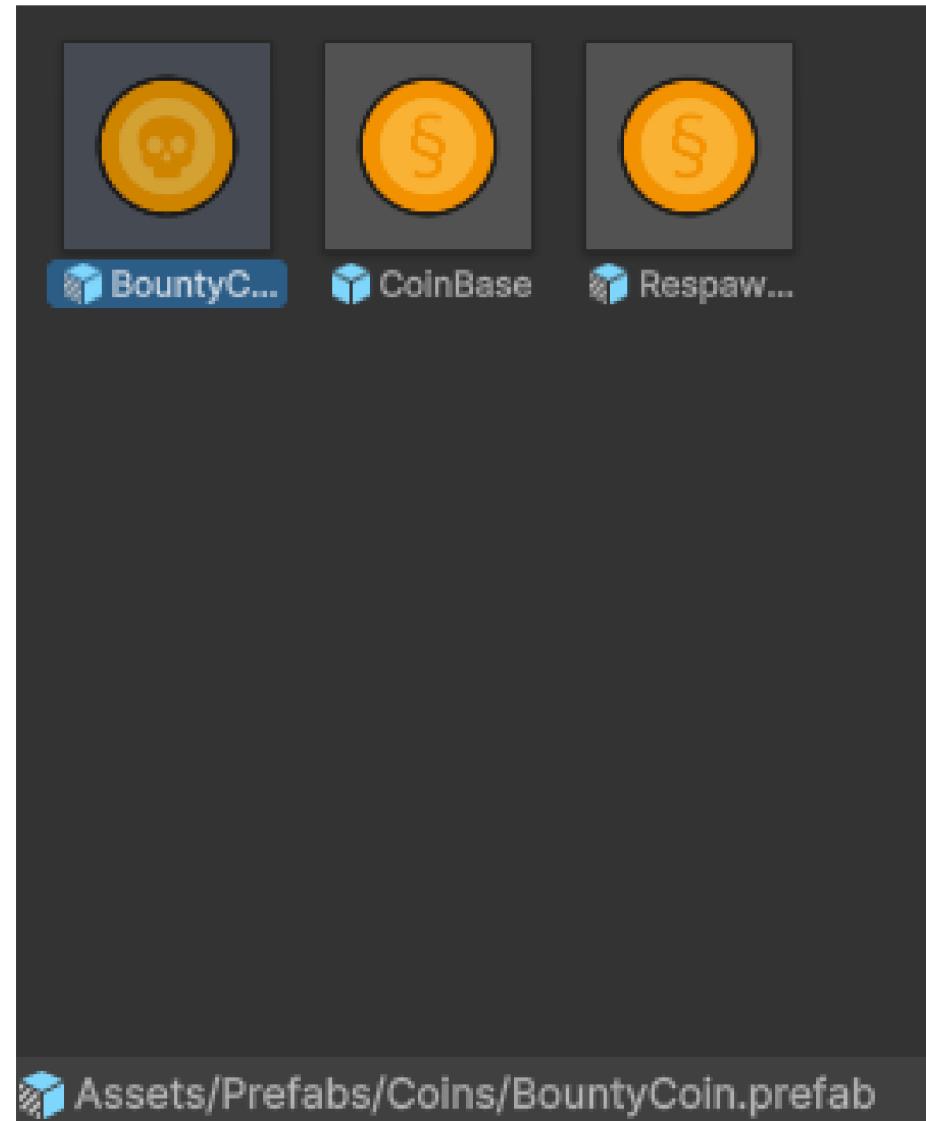
Assets > Prefabs > Coins



CoinBase



Respawn...



Assets/Scripts/Core/Coins/BountyCoin.cs

Bounty Coin

- Inherit from **Coin**
- Copy logic from **RespawningCoin** for Collect method
- Destroy the coin instead of invoking an event since we aren't needing to respawn this coin elsewhere

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

④ Unity Script | 0 references
public class BountyCoin : Coin
{
    2 references
    public override int Collect()
    {
        throw new System.NotImplementedException();
    }
}
```

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

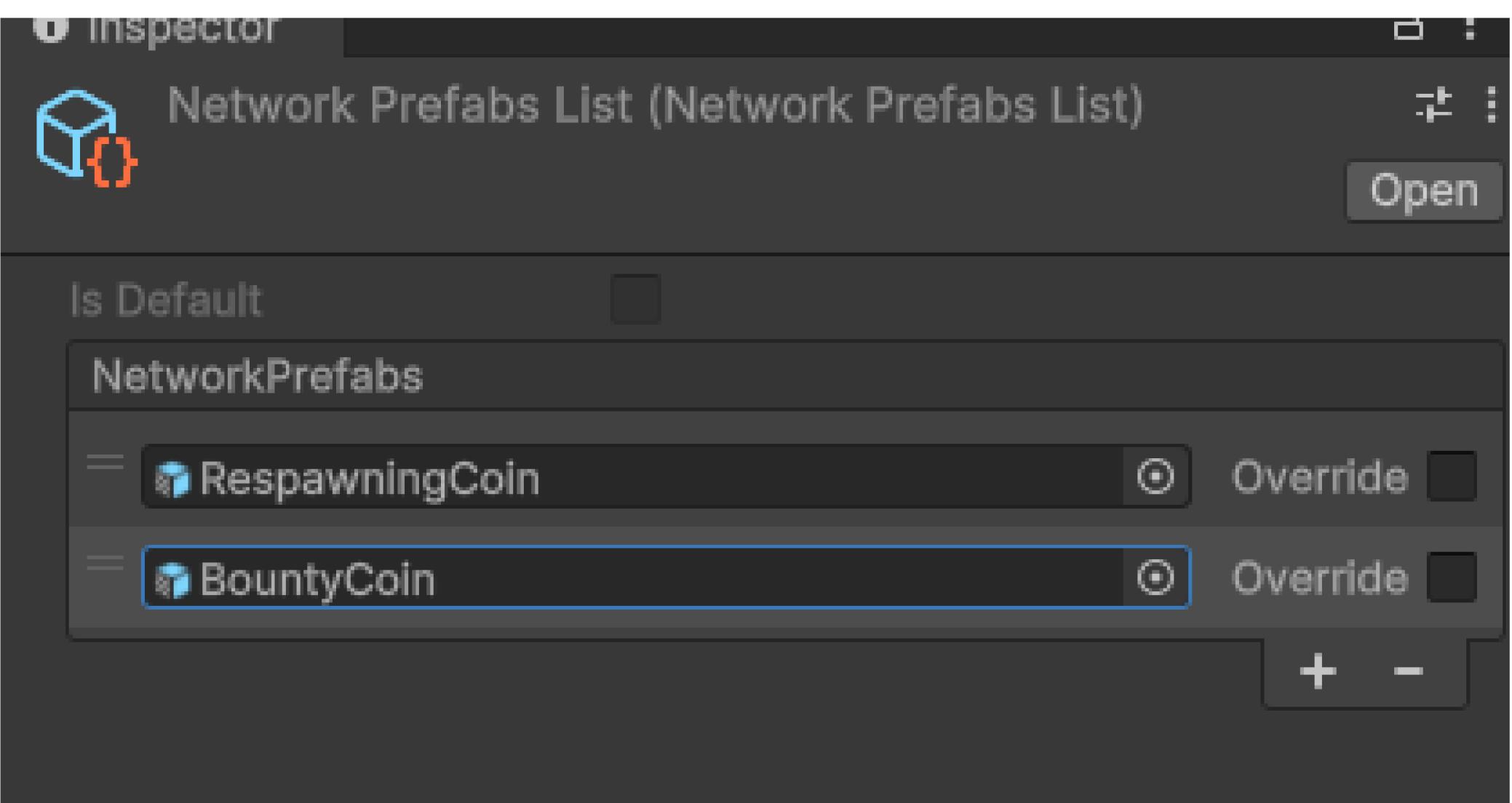
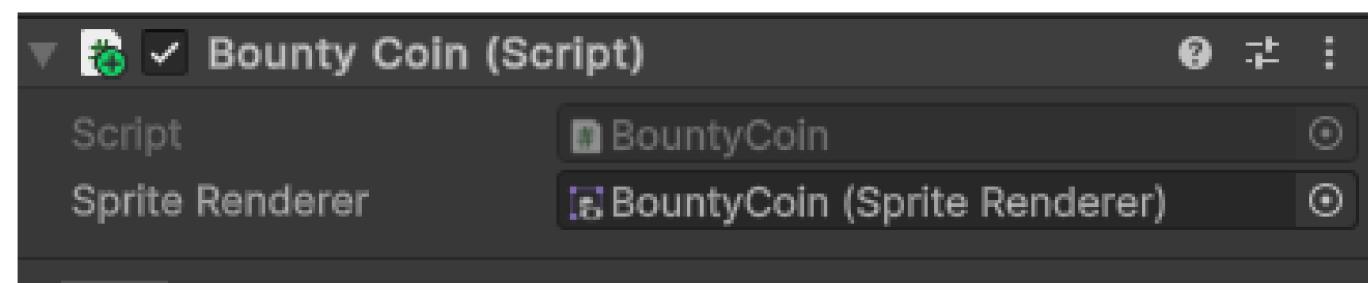
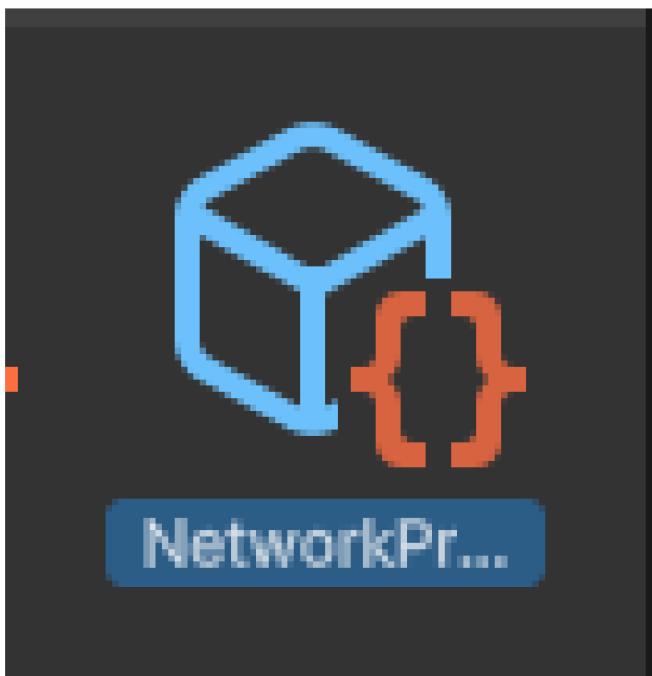
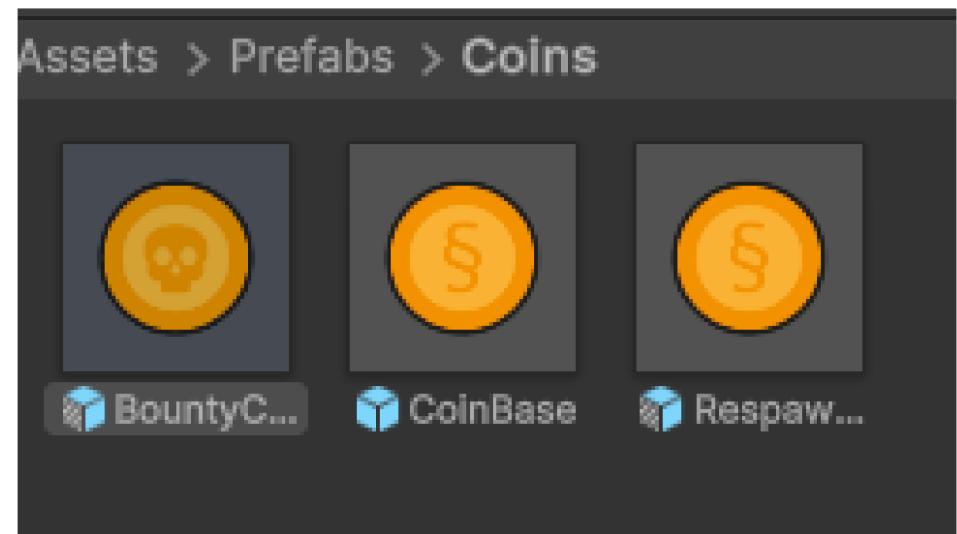
④ Unity Script | 0 references
public class BountyCoin : Coin
{
    2 references
    public override int Collect()
    {
        if (!IsServer)
        {
            Show(false);
            return 0;
        }

        if (alreadyCollected) { return 0; }

        alreadyCollected = true;

        Destroy(gameObject);

        return coinValue;
    }
}
```



C# CoinWallet.cs

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using Unity.Netcode;
5  using UnityEngine;
6
7  public class CoinWallet : NetworkBehaviour
8  {
9    [Header("References")]
10   [SerializeField] private Health health;
11   [SerializeField] private BountyCoin coinPrefab;
12
13  [Header("Settings")]
14  [SerializeField] private float coinSpread = 3f;
15  [SerializeField] private int bountyCoinCount = 10;
16  [SerializeField] private int minBountyCoinValue = 5;
17
18  public NetworkVariable<int> TotalCoins = new NetworkVariable<int>();
19}
```

ໃຫ້ກຳກົດ Copy ຈາກໃນ CoinSpawner ໃຈໃສ່ໃນ CoinWallet

Unity Script (1 asset reference) | 0 references

```
6  public class CoinSpawner : NetworkBehaviour
7  {
8      [SerializeField] private RespawningCoin coinPrefab;
9      [SerializeField] private int maxCoins = 50;
10     [SerializeField] private int coinValue = 10;
11     [SerializeField] private Vector2 xSpawnRange;
12     [SerializeField] private Vector2 ySpawnRange;
13     [SerializeField] private LayerMask layerMask;
14
15     private Collider2D[] coinBuffer = new Collider2D[1];
16
17     private float coinRadius;
```



```
class CoinWallet : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private Health health;
    [SerializeField] private BountyCoin coinPrefab;

    [Header("Settings")]
    [SerializeField] private float coinSpread = 3f;
    [SerializeField] private int bountyCoinCount = 10;
    [SerializeField] private int minBountyCoinValue = 5;
    [SerializeField] private LayerMask layerMask;

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

    private float coinRadius;

    public NetworkVariable<int> TotalCoins = new NetworkVariable<int>();
```

C# CoinWallet.cs

```
24
25     0 references
26     public override void OnNetworkSpawn()
27     {
28         if (!IsServer) { return; }
29         coinRadius = coinPrefab.GetComponent<CircleCollider2D>().radius;
30         health.OnDie += HandleDie;
31     }
32     0 references
33     public override void OnNetworkDespawn()
34     {
35         if(!IsServer) { return; }
36         health.OnDie -= HandleDie;
}
```

```
private void HandleDie(Health health)
{
    ...
}
```

⊕ Unity Script (1 asset reference) | 2 references

```
public class CoinWallet : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private Health health;
    [SerializeField] private BountyCoin coinPrefab;

    [Header("Settings")]
    [SerializeField] private float coinSpread = 3f;
    [SerializeField] private float bountyPercentage = 50f;
    [SerializeField] private int bountyCoinCount = 10;
    [SerializeField] private int minBountyCoinValue = 5;
    [SerializeField] private LayerMask layerMask;

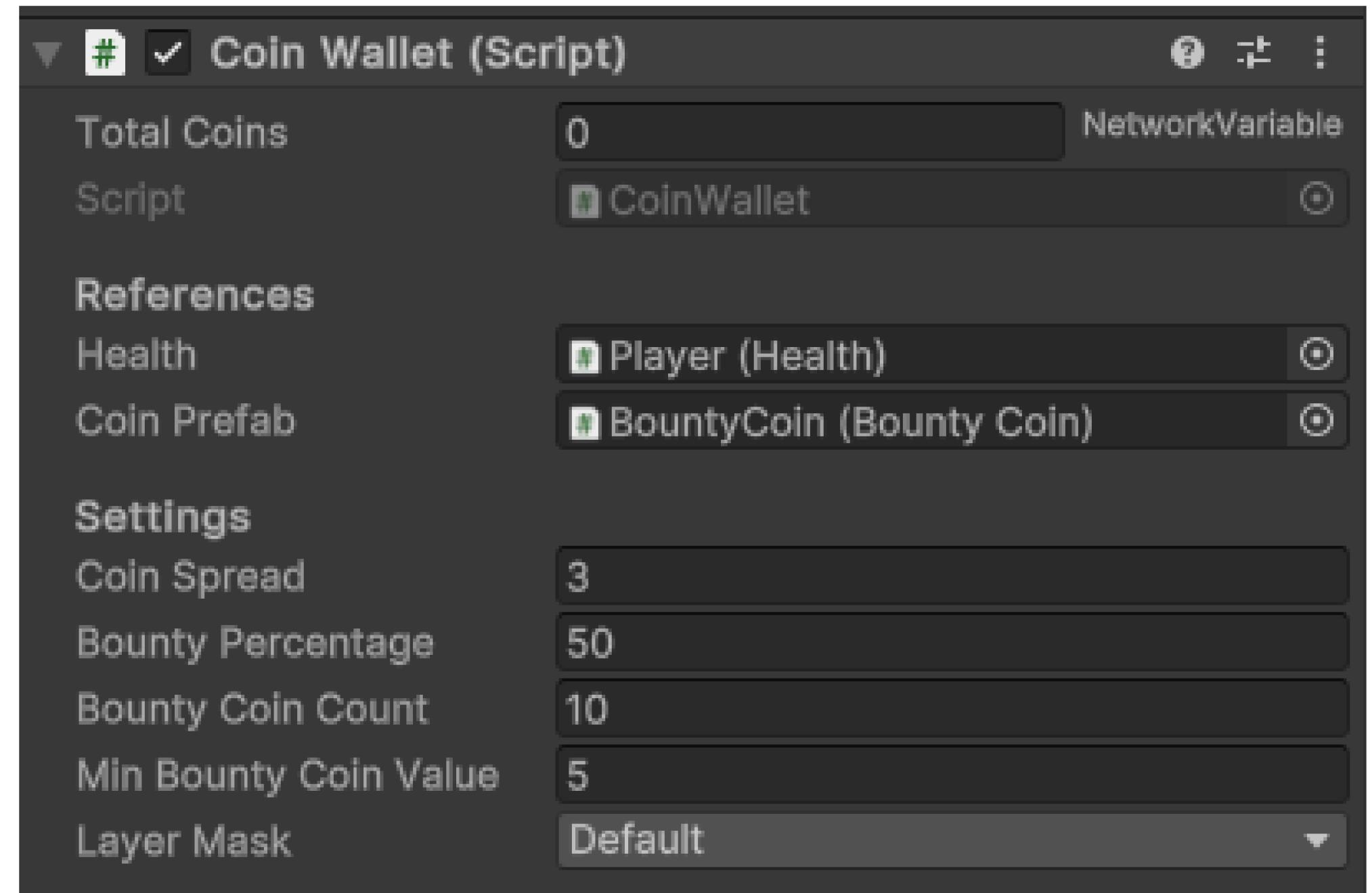
    private Collider2D[] coinBuffer = new Collider2D[1];

    private float coinRadius;
```

เราจะทำการ Copy GetSpawnPoint จากใน CoinSpawner มาใส่ใน CoinWallet และปรับค่าบางส่วน

```
2 references
private void HandleDie(Health health)
{
    int bountyValue = (int)(TotalCoins.Value * (bountyPercentage / 100f));
    int bountyCoinValue = bountyValue / bountyCoinCount;
    if (bountyCoinValue < minBountyCoinValue) { return; }

    for (int i = 0; i < bountyCoinCount; i++)
    {
        BountyCoin coinInstance = Instantiate(coinPrefab, GetSpawnPoint(), Quaternion.identity);
        coinInstance.SetValue(bountyCoinValue);
        coinInstance.NetworkObject.Spawn();
    }
}
1 reference
private Vector2 GetSpawnPoint()
{
    while (true)
    {
        Vector2 spawnPoint = (Vector2)transform.position + UnityEngine.Random.insideUnitCircle * coinSpread;
        ContactFilter2D contactFilter2D = new ContactFilter2D();
        contactFilter2D.layerMask = layerMask;
        int numColliders = Physics2D.OverlapCircle(spawnPoint, coinRadius, contactFilter2D, coinBuffer);
        if(numColliders == 0)
        {
            return spawnPoint;
        }
    }
}
```





Settings	
Coin Spread	3
Bounty Percentage	50
Bounty Coin Count	10
Min Bounty Coin Value	5

ถ้าค่าแบบเพ้อไปก็ให้ปรับจูนใน Setting ของ CoinWallet

Assignment

ให้ทำการด่ายคลิปผลลัพธ์ของการทำ Workshop ตามเนื้อหา Workshop ໃນແຕ່ລະ
หัวข้อนี้พร้อมອธิบายประกอบ

- Leaderboard Names
- Leaderboard Updating
- Leaderboard Sorting
- Losing Coins
- Bounty Coins



<https://discord.gg/24xmUFHzR>

WOLVEDEN ACADEMY



facebook.com/GameDevMew