

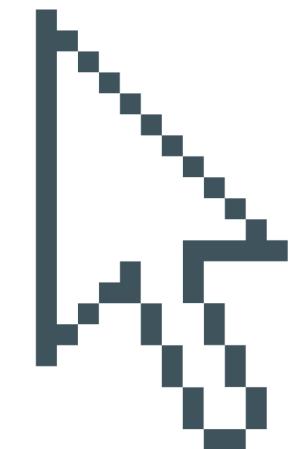


WOLVEDEN ACADEMY

NETWORKING AND MULTIPLAYER ONLINE GAMES



# MULTIPLAYER NEW ERA



BY PONGSATHORN KIATTICHAOENPORN (MEW)

WEEK 9 : GAMEPLAY ADDITIONS 1

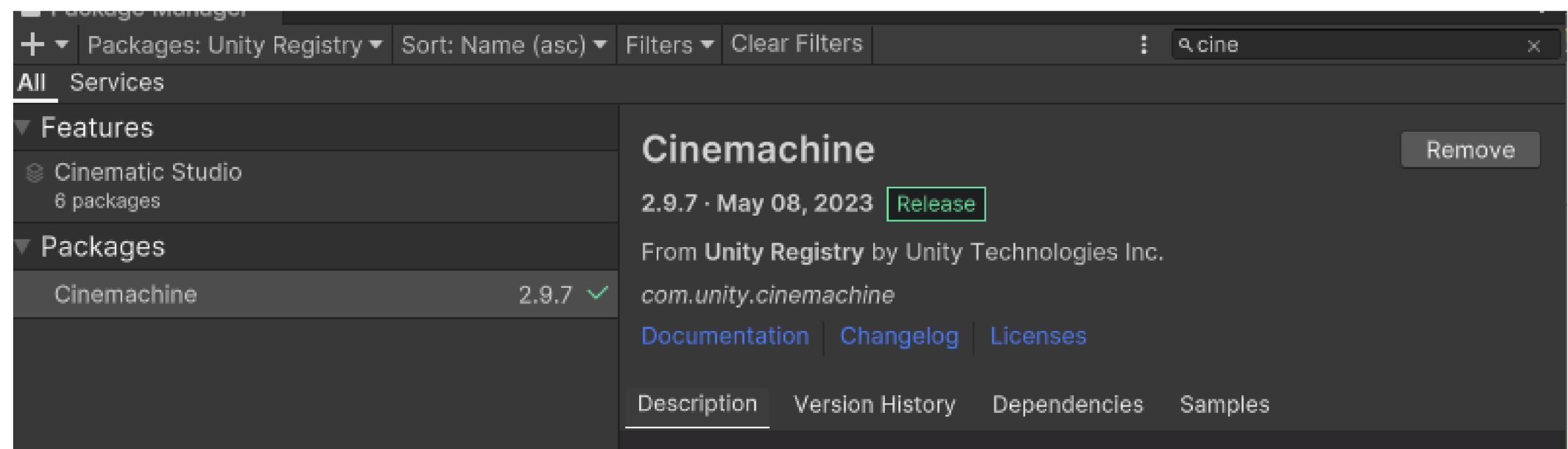
# **Gameplay Additions 1**

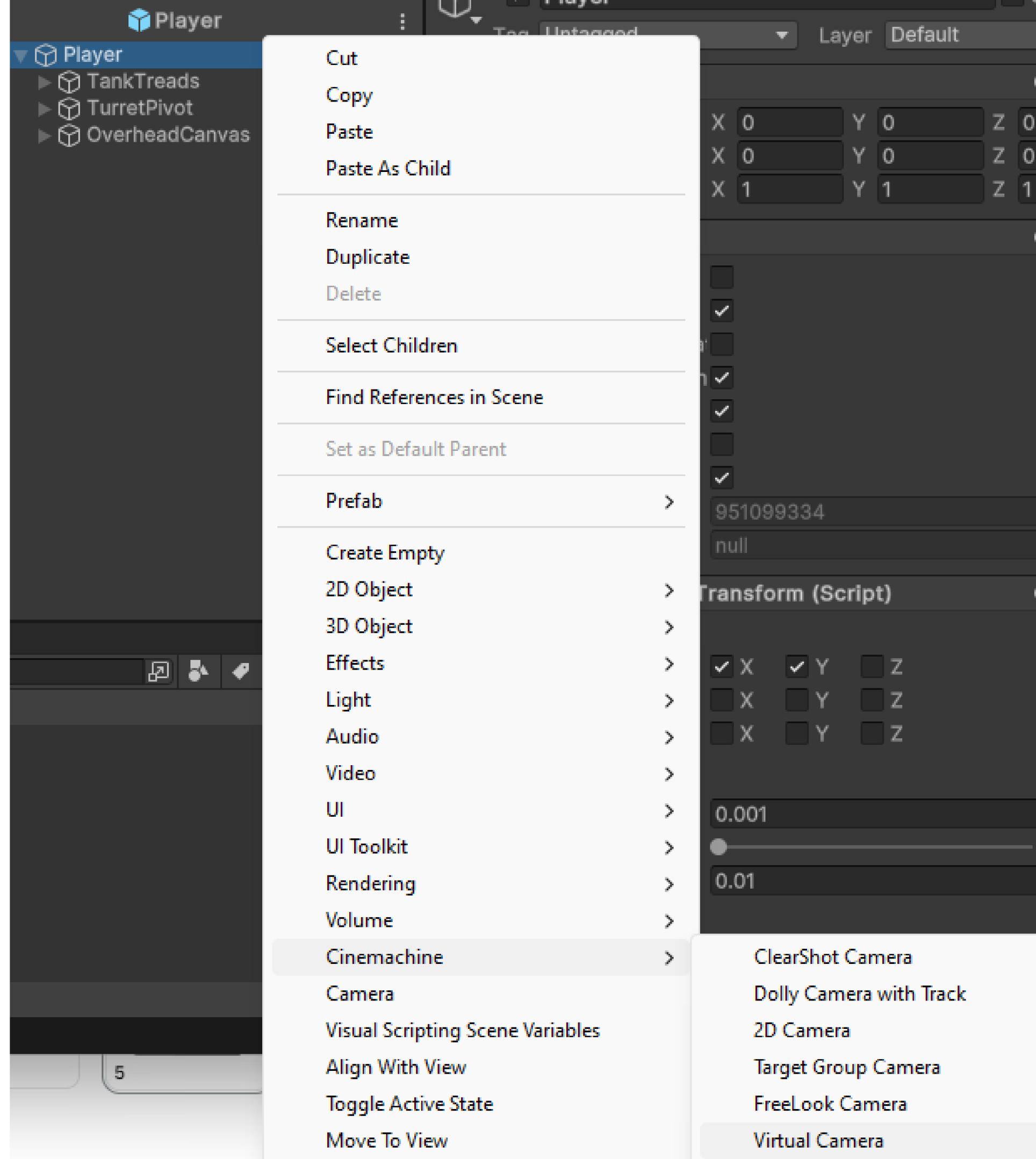
# Topic in this week

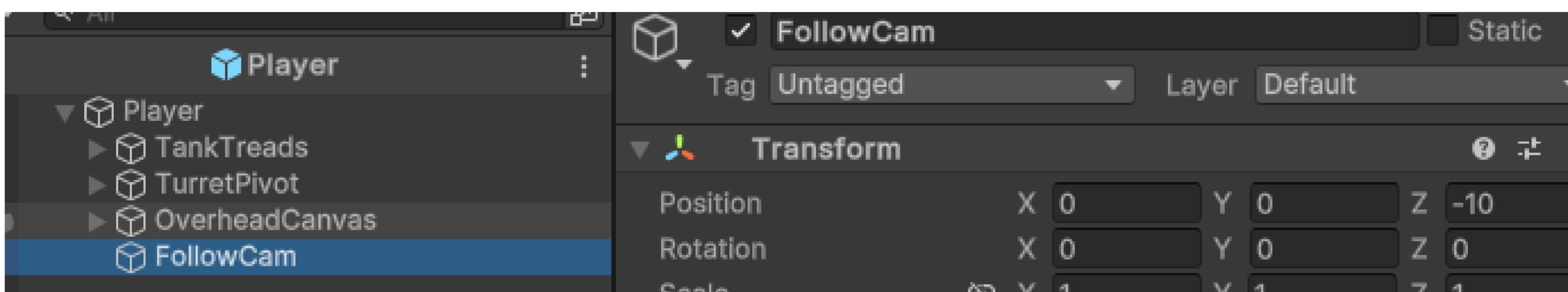
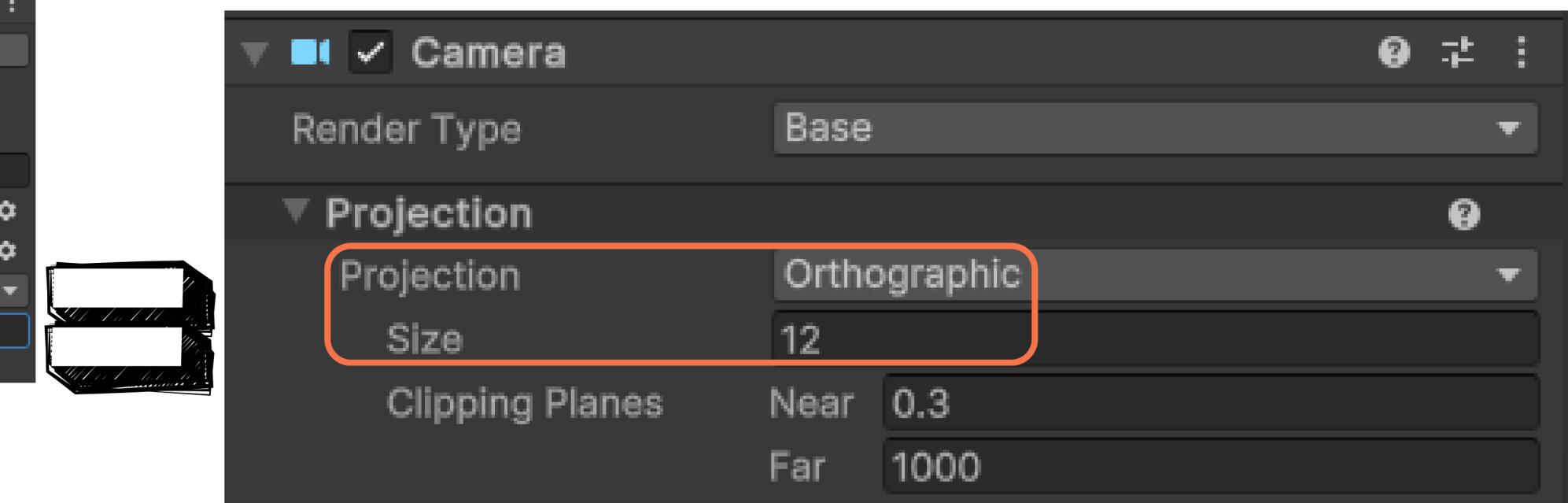
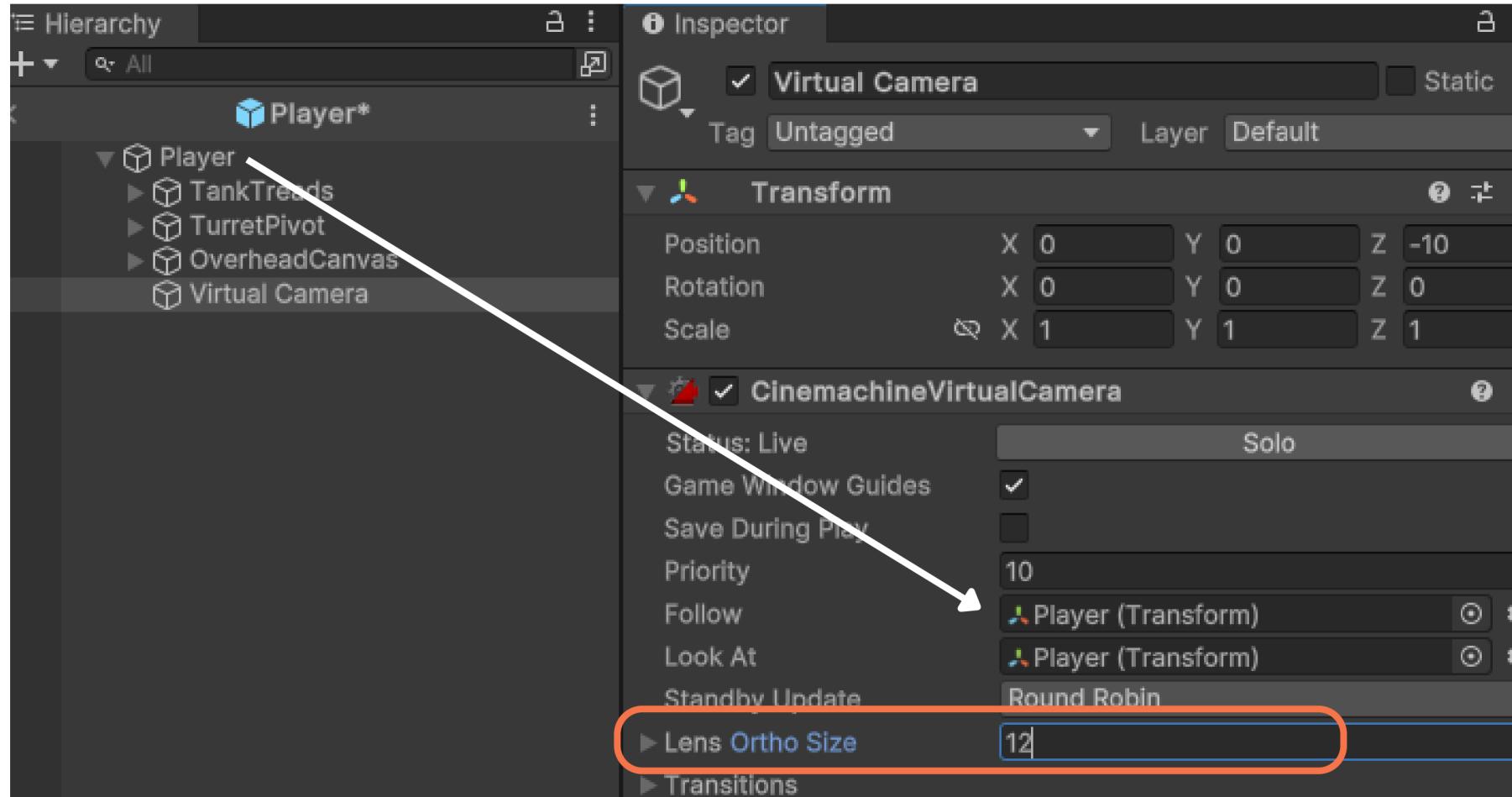
## Workshop

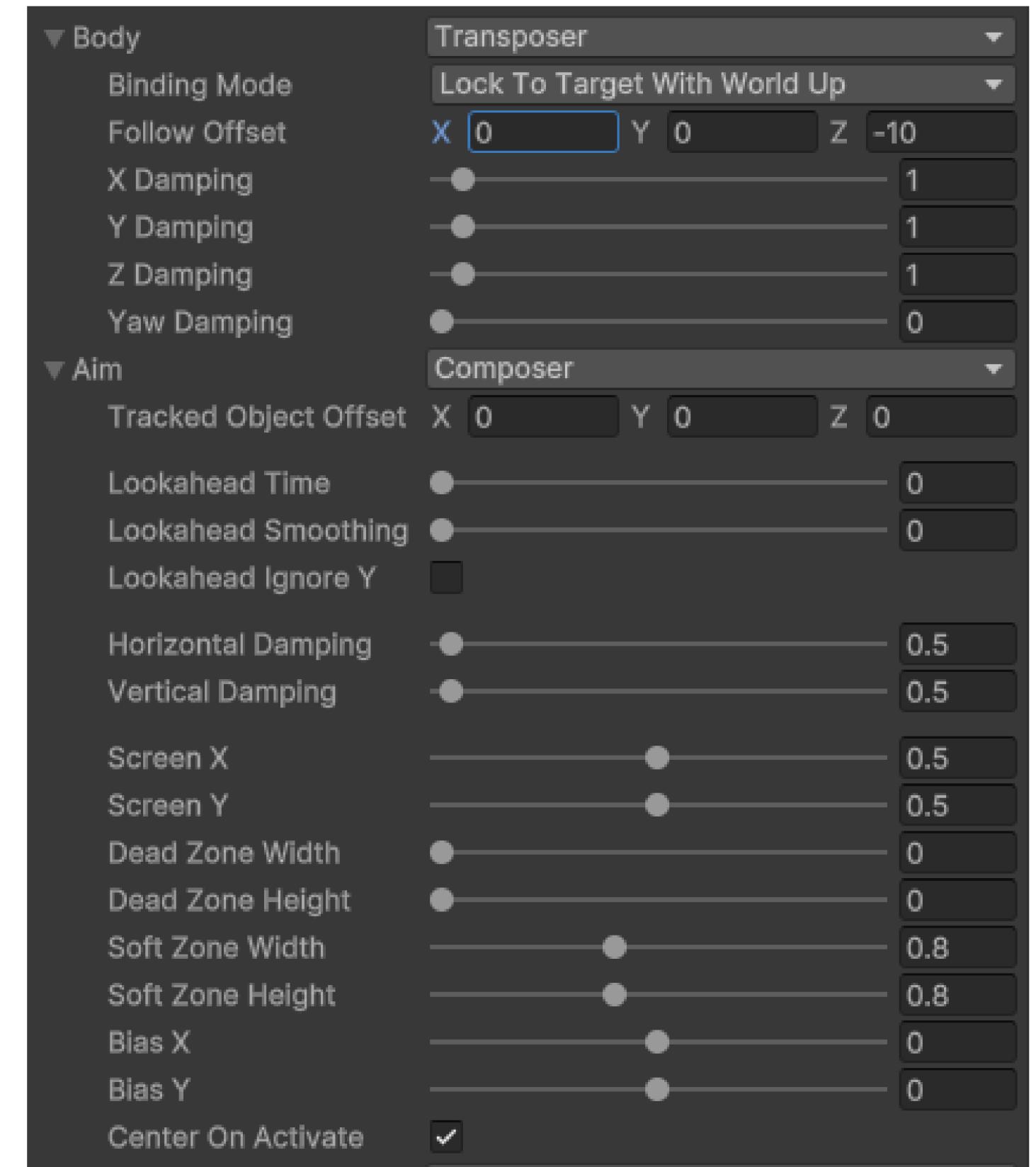
- Player Camera
- Overhead Names
- Change Color
- Spawn Points
- Respawning

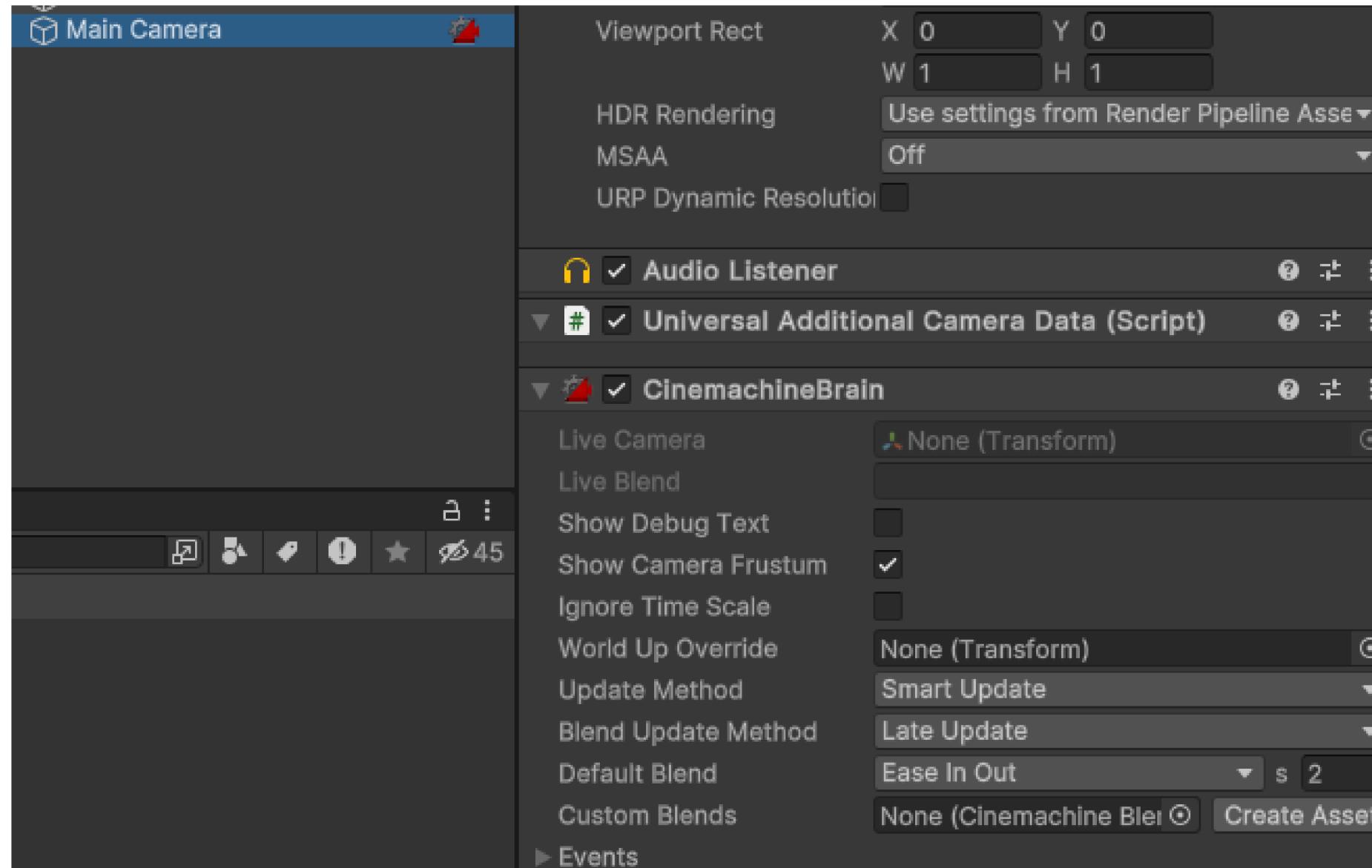
# **Player Camera**



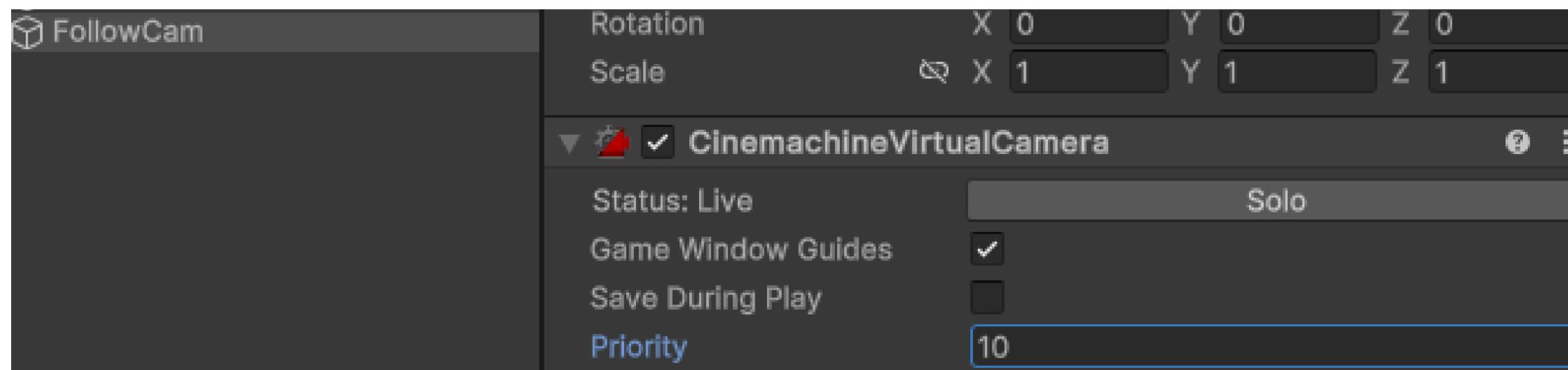






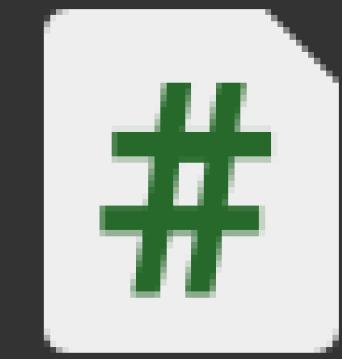


ถ้ากดสوبตอนนี้เราจะเห็นว่ากล้องสามารถติดตามตัวละครได้แล้ว...



เพียงแต่ว่า Priority ของกล้องนั้นจะซ้ำซ้อนกันในทุก player ทำให้ตัวเกมจะมีความสับสนได้ว่าจะตามที่กล้องไหนกันแน่ ดังนั้น เราต้องมีการเขียน Script เพื่อจัดการในส่วนนี้

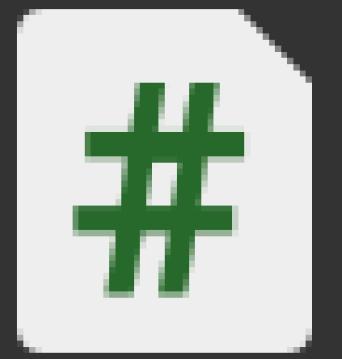
Assets > Scripts > Core > Player



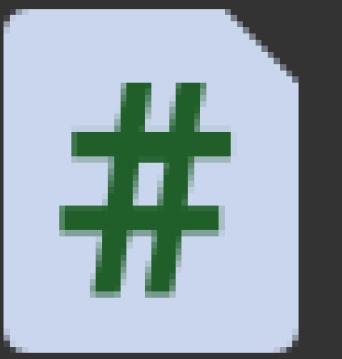
PlayerAiming



PlayerMov...



ProjectileLa...



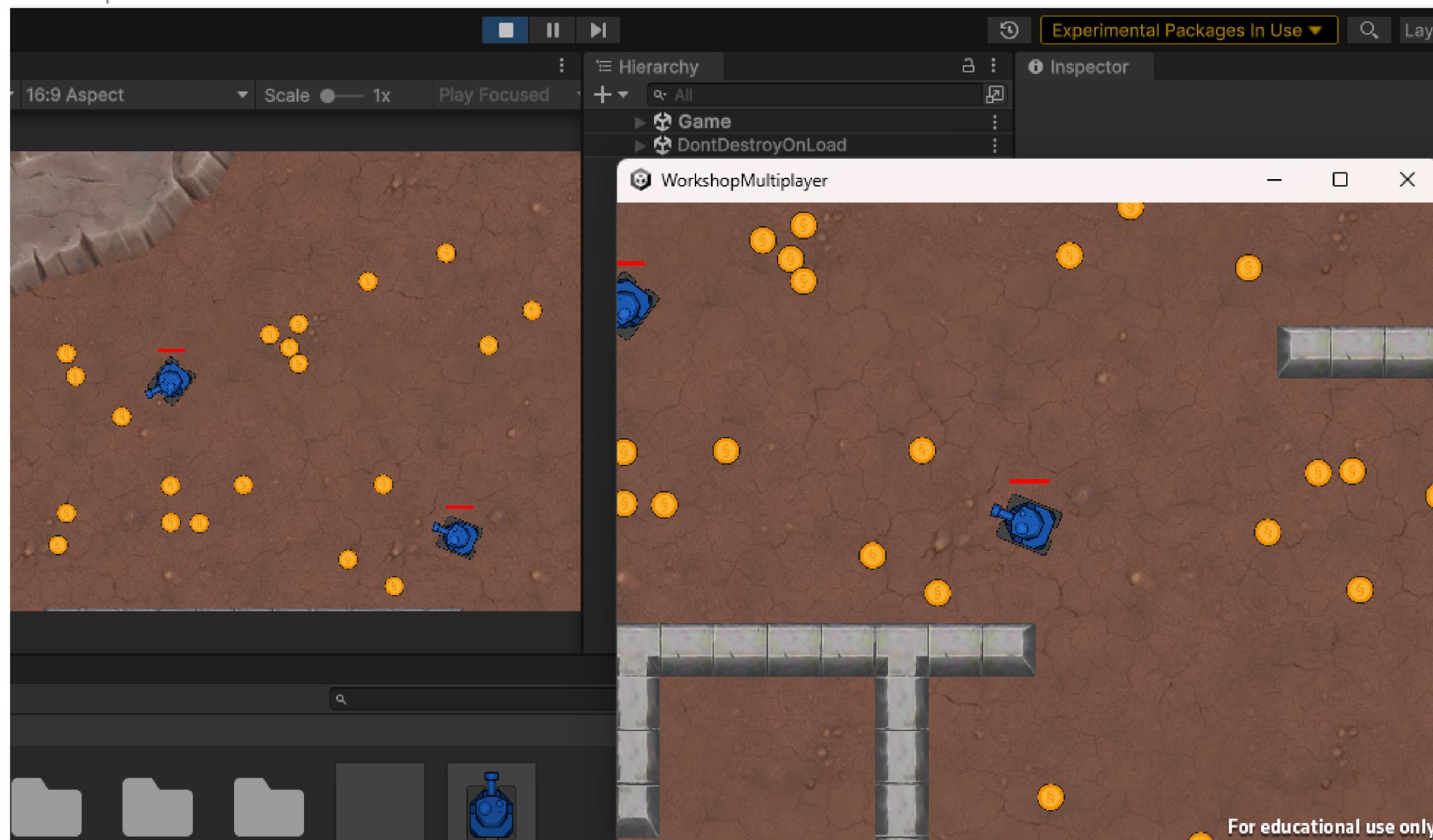
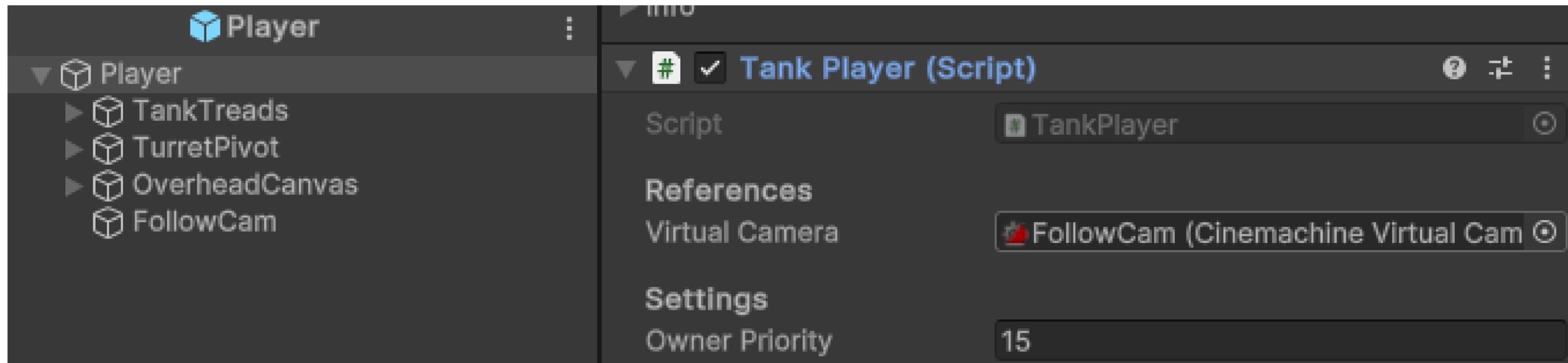
TankPlayer

```
1  ┌─ using System.Collections;
2  │  using System.Collections.Generic;
3  │  using Unity.Netcode;
4  │  using UnityEngine;
5  │  using Cinemachine;
6
7  ┌─ Unity Script | 0 references
8  ┌─ public class TankPlayer : NetworkBehaviour
9  {
10
11    ┌─ public override void OnNetworkSpawn()
12    {
13      ┌─ if (IsOwner)
14      {
15        ┌─
16      }
17    }
18 }
```

# Camera Priority

- Expose a field reference for the player's  
**CinemachineVirtualCamera**
- Expose a field value **int** for the camera priority if you are the owner
- When the object is spawned on the network, set the camera's  
**Priority** if we are the owner

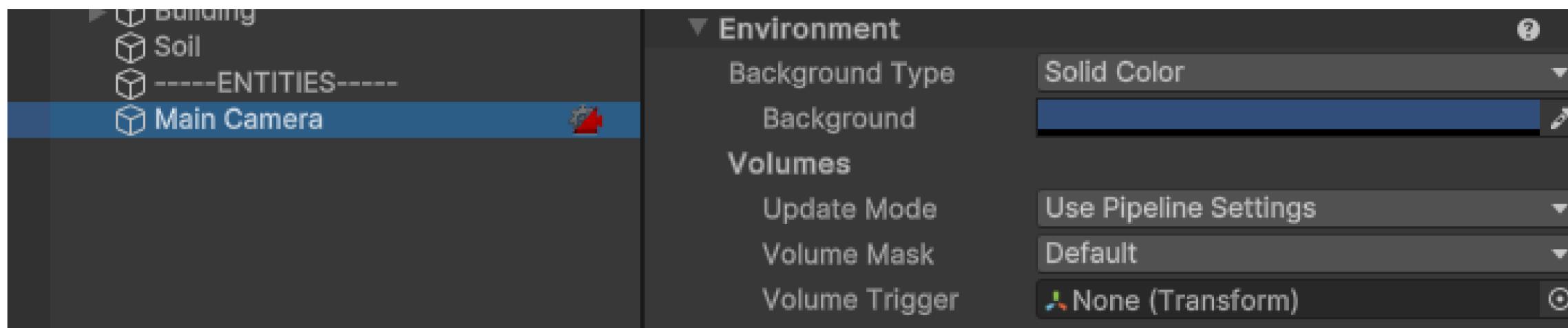
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Unity.Netcode;
4  using UnityEngine;
5  using Cinemachine;
6
7  public class TankPlayer : NetworkBehaviour
8  {
9      [Header("References")]
10     [SerializeField] private CinemachineVirtualCamera virtualCamera;
11
12     [Header("Settings")]
13     [SerializeField] private int ownerPriority = 15;
14
15     public override void OnNetworkSpawn()
16     {
17         if (IsOwner)
18         {
19             virtualCamera.Priority = ownerPriority;
20         }
21     }
}
```



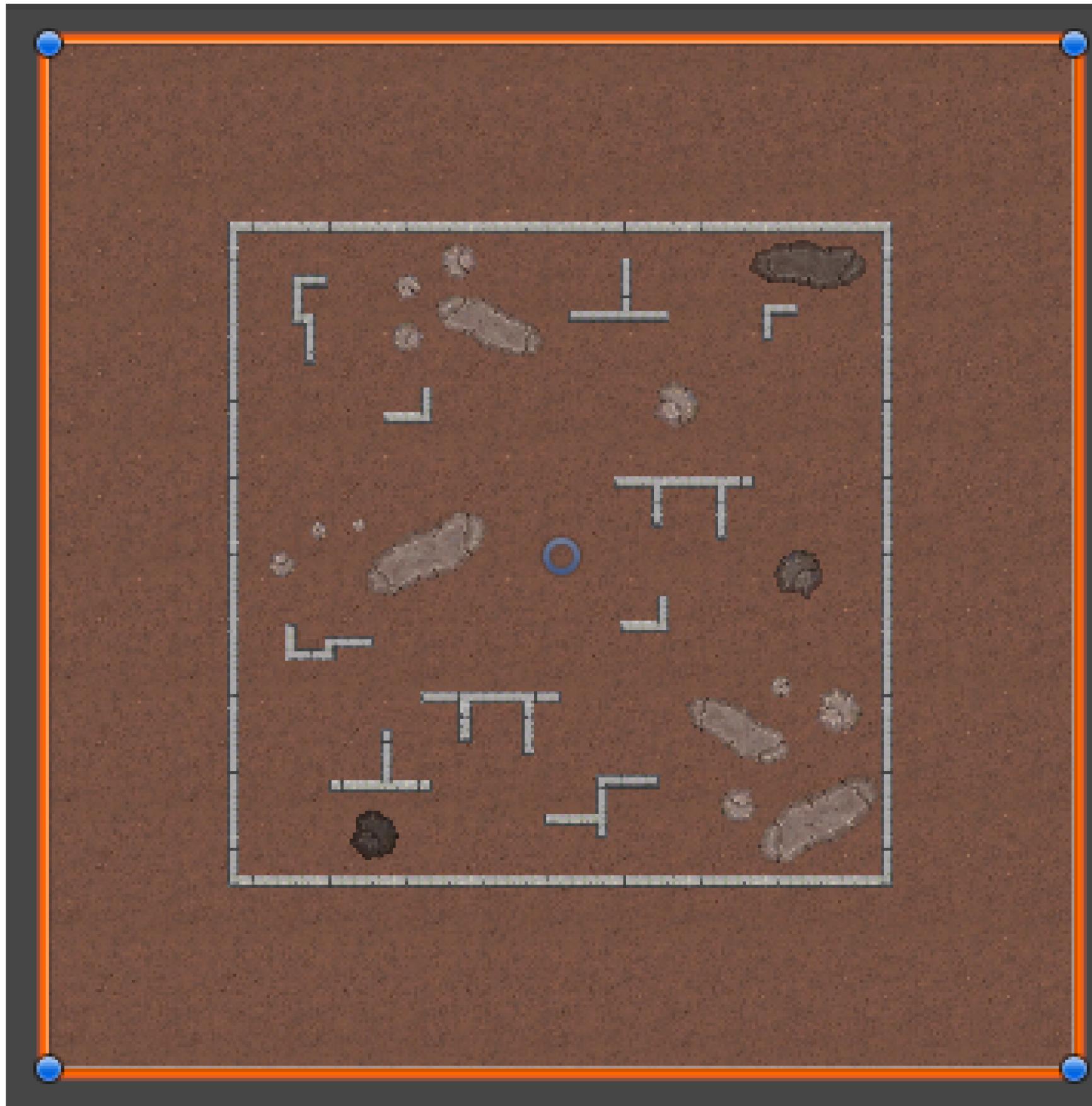
ตอนวิ่งมาดึงขอบจะสั้นเกตเဟนได้ว่าเป็น BG สีฟ้า



สามารถเปลี่ยนสีโดยแก้ที่ส่วนนี้



# หรือจะใช้วิธีขยายพื้นที่ให้ใหญ่เกินขอบมากขึ้นก็ได้



Game\*

- SYSTEMS-----
- CoinSpawner
- ENVIRONMENT-----
- Global Light 2D
- Rock
- Building
- Soil
- ENTITIES-----
- Main Camera

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Sprite Renderer

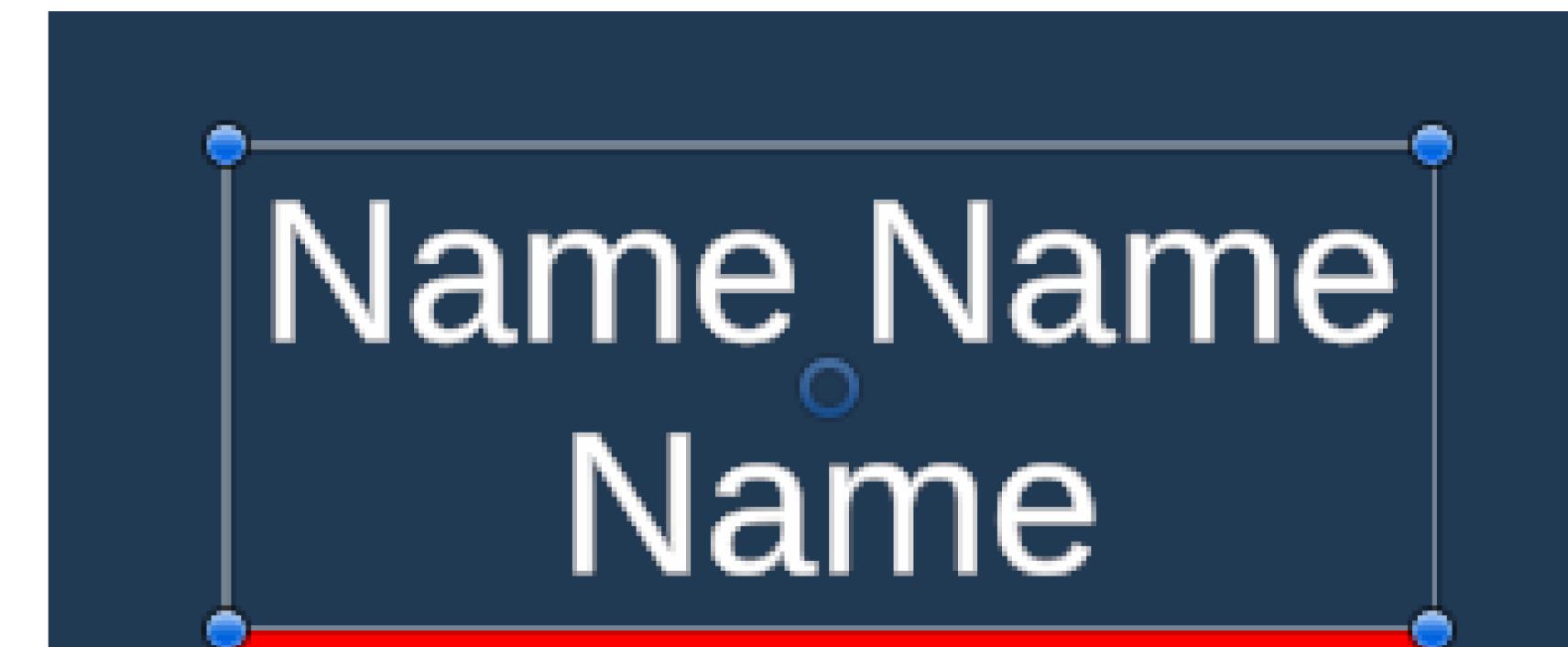
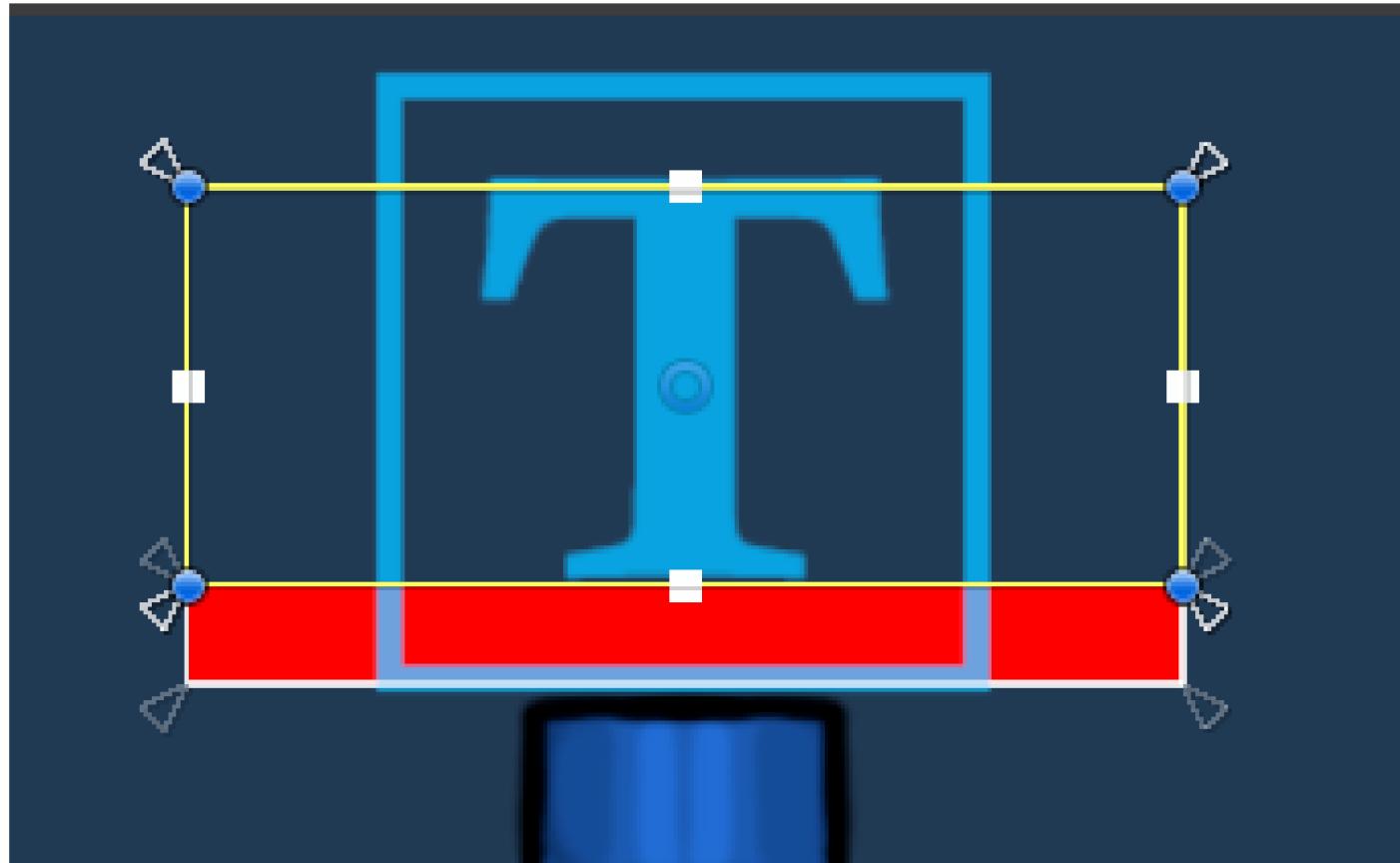
Sprite	Martian_Soil1
Open Sprite Editor	
Color	[Color Swatch]
Flip	<input type="checkbox"/> X <input type="checkbox"/> Y
Draw Mode	Tiled

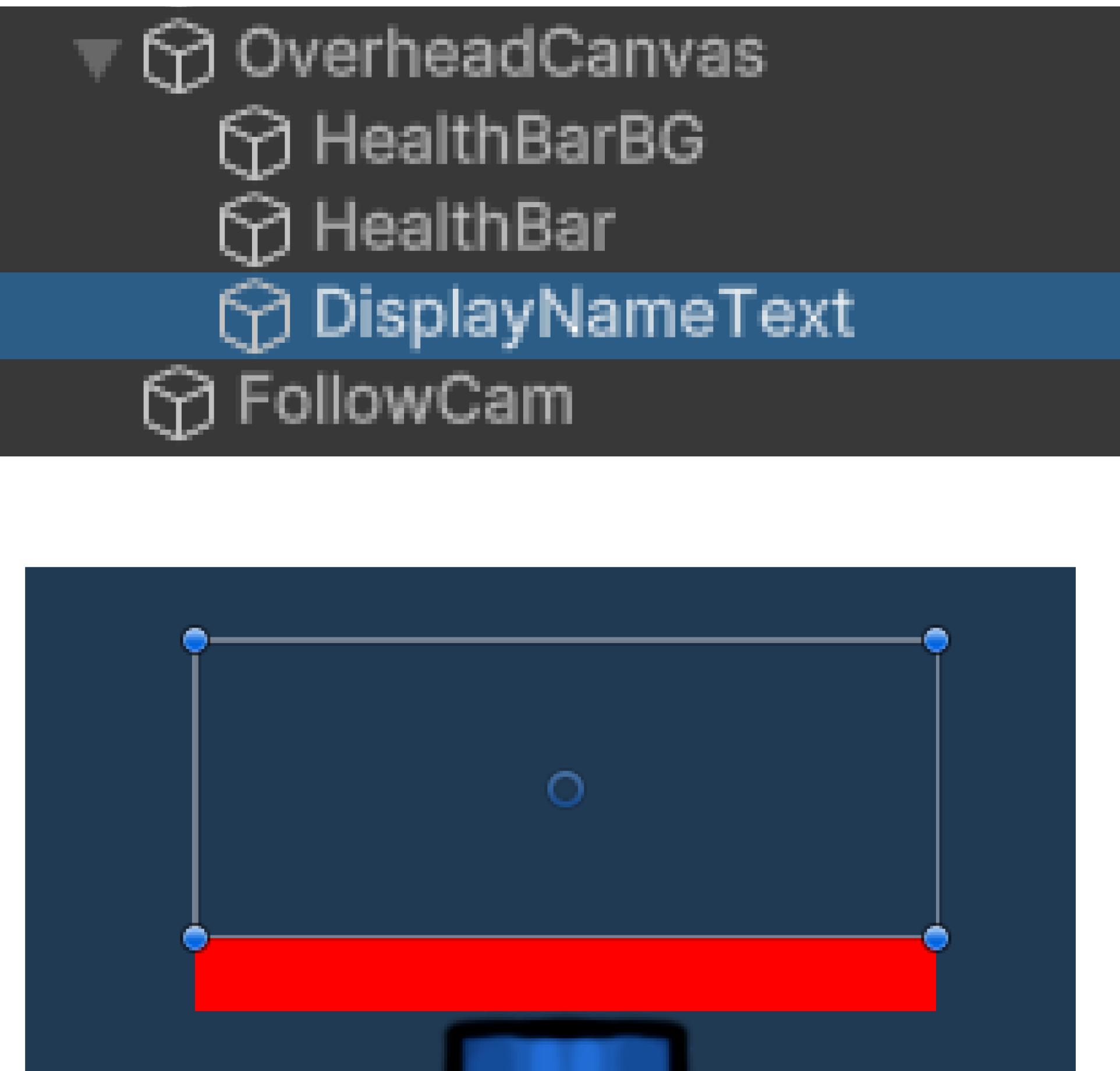
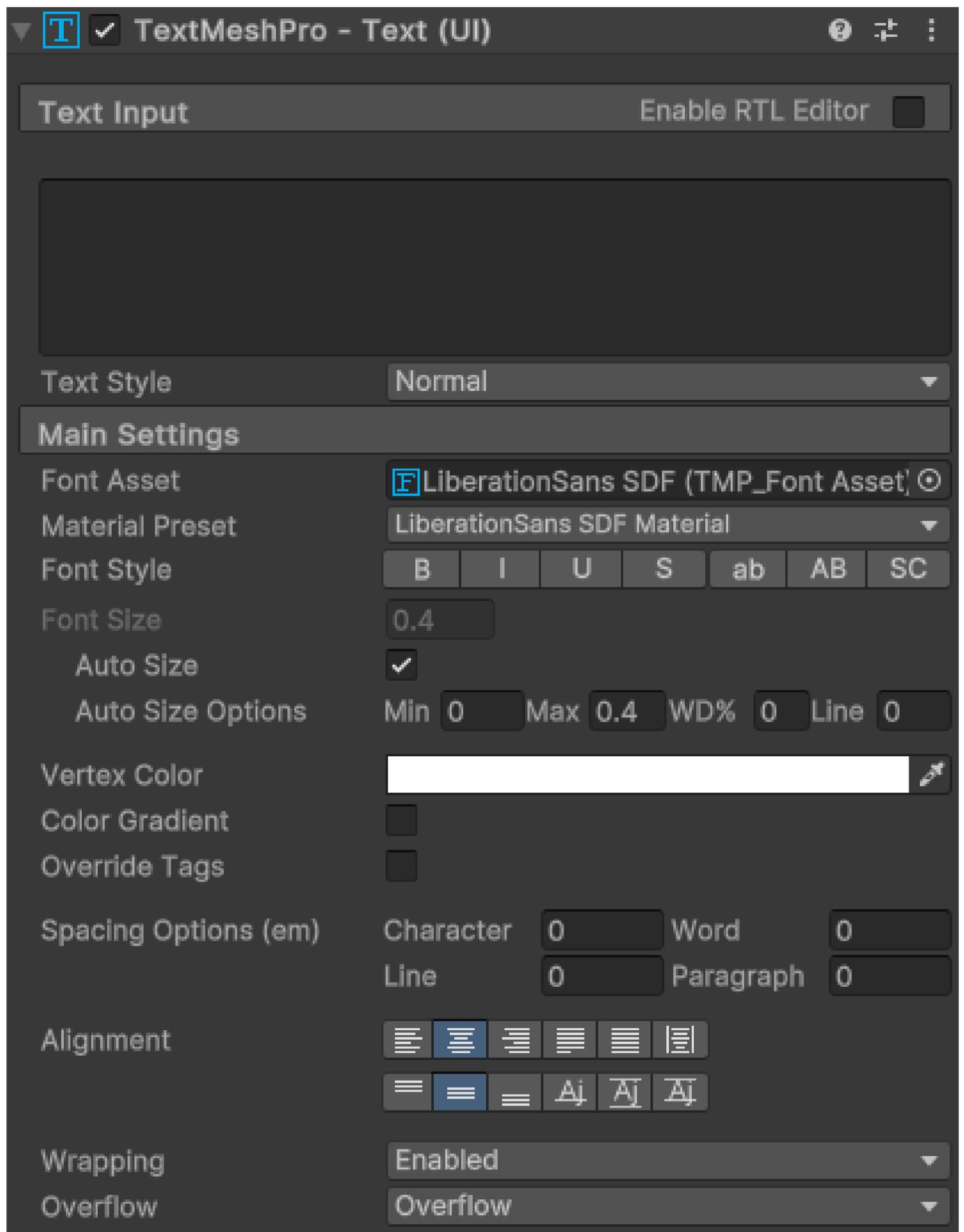
**Warning:** Sprite Tiling might not appear correctly because the Sprite used is not generated with Full Rect or Sprite Mode is set to Polygon mode. To fix this, change the Mesh Type in the Sprite's Import setting to Full Rect and Sprite Mode is either Single or Multiple.

Size	Width	Height
160	160	

Tile Mode: Continuous  
Mask Interaction: None  
Sprite Sort Point: Center  
Material: Sprite-Lit-Default

# **Overhead Names**





```
private Dictionary<ulong, string> clientIdToAuth = new Dictionary<ulong, string>();
private Dictionary<string, UserData> authIdToUserData = new Dictionary<string, UserData>();
```

ถึงเวลาที่เราจะได้ใช้ UserData แล้ว เพื่อดึงข้อมูลซึ่งออกมา

โดยจะใช้วิธีดึงข้อมูลจาก Dictionary โดยใช้รูปแบบ ตามตัวอย่างด้านล่างนี้

```
if(authIdToUserData.TryGetValue("something",out UserData data))
{
}
}
```

# Get User Data

- Create a **public** method that returns **UserData** and takes in a **ulong clientId**
- Use TryGetValue on the **clientIdToAuth** dictionary
- If successful, use **TryGetValue** on the **authIdToUserData** dictionary and return the **UserData**, otherwise return **null**

## C# NetworkServer.cs

```
0 references
51     public UserData GetUserDataByClientId(ulong clientId)
52     {
53         if(clientIdToAuth.TryGetValue(clientId,out string authId))
54         {
55             if(authIdToUserData.TryGetValue(authId,out UserData data))
56             {
57                 return data;
58             }
59             return null;
60         }
61         return null;
62     }
```

```
18     public class HostGameManager : IDisposable
19     {
20         private Allocation allocation;
21         private string joinCode;
22         private string lobbyId;
23
24         private NetworkServer networkServer;
25
26         private const int MaxConnections = 20;
27         private const string GameSceneName = "Game";
    1 reference
```



```
  REFERENCES
18     public class HostGameManager : IDisposable
19     {
20         private Allocation allocation;
21         private string joinCode;
22         private string lobbyId;
23
24
25
26
27
    0 references
    public NetworkServer NetworkServer { get; private set; }

    private const int MaxConnections = 20;
    private const string GameSceneName = "Game";
    1 reference
```

```
  1 reference
18     public async void Dispose()
19     {
20         HostSingleton.Instance.StopCoroutine(nameof(HeartbeatLobby));
21
22         if(!string.IsNullOrEmpty(lobbyId) )
23         {
24             try
25             {
26                 await Lobbies.Instance.DeleteLobbyAsync(lobbyId);
27             }
28             catch(LobbyServiceException e)
29             {
30                 Debug.Log(e);
31             }
32
33             lobbyId = string.Empty;
34         }
35
36         NetworkServer?.Dispose();
37     }
```

```
  81
  82     NetworkServer = new NetworkServer(NetworkManager.Singleton);
  83
```

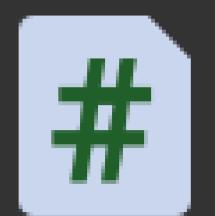
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Unity.Netcode;
4  using UnityEngine;
5  using Cinemachine;
6  using Unity.Collections;
7
8  public class TankPlayer : NetworkBehaviour
9  {
10     [Header("References")]
11     [SerializeField] private CinemachineVirtualCamera virtualCamera;
12
13     [Header("Settings")]
14     [SerializeField] private int ownerPriority = 15;
15
16     public NetworkVariable<FixedString32Bytes> PlayerName = new NetworkVariable<FixedString32Bytes>();
17
18     public override void OnNetworkSpawn()
19     {
20         if (IsServer)
21         {
22             UserData userData =
23                 HostSingleton.Instance.GameManager.NetworkServer.GetUserDataByClientId(OwnerId);
24             PlayerName.Value = userData.userName;
25         }
26
27         if (IsOwner)
28         {
29             virtualCamera.Priority = ownerPriority;
30         }
31     }
32 }
```



PlayerAiming



PlayerMov...



PlayerName...



ProjectileLa...



TankPlayer

## # Assets/Scripts/Core/Player/PlayerNameDisplay.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using TMPro;
4  using UnityEngine;
5
6  public class PlayerNameDisplay : MonoBehaviour
7  {
8      [SerializeField] private TankPlayer player;
9      [SerializeField] private TMP_Text playerNameText;
10 }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using TMPro;
5  using Unity.Collections;
6  using UnityEngine;
7
8  public class PlayerNameDisplay : MonoBehaviour
9  {
10     [SerializeField] private TankPlayer player;
11     [SerializeField] private TMP_Text playerNameText;
12
13     void Start()
14     {
15         HandlePlayerNameChanged(string.Empty, player.PlayerName.Value);
16
17         player.PlayerName.OnValueChanged += HandlePlayerNameChanged;
18     }
19
20     private void HandlePlayerNameChanged(FixedString32Bytes oldName, FixedString32Bytes newName)
21     {
22         playerNameText.text = newName.ToString();
23     }
24
25     void OnDestroy()
26     {
27         player.PlayerName.OnValueChanged -= HandlePlayerNameChanged;
28     }
29 }
```

Player

1.5 0.75 L R

Player

TankTreads

TurretPivot

OverheadCanvas

HealthBarBG

HealthBar

DisplayNameText

FollowCam

Anchor

Min X 0 Y 0

Max X 0 Y 0

Pivot X 0.5 Y 0.5

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Canvas

Render Mode World Space

Event Camera None (Camera)

A World Space Canvas with no specified Event Camera may not register UI events correctly.

Sorting Layer Player

Order in Layer 150

Additional Shader Channel TexCoord1, Normal, Tangent

Vertex Color Always In G

Health Display (Script)

Script HealthDisplay

References

Health Player (Health)

Health Bar Image HealthBar (Image)

Player Name Display (Script)

Script PlayerNameDisplay

Player Player (Tank Player)

Player Name Text DisplayNameText (Text Mesh Pro U)



Display Name Text

Static

Tag Untagged Layer Default

Rect Transform

stretch

Left: -0.5 Top: -0.25 Pos Z: 0  
Right: Bottom: 0

custom

Anchors

Min X: 0 Y: 0.2  
Max X: 1 Y: 1  
Pivot X: 0.5 Y: 0.5

Rotation X: 0 Y: 0 Z: 0  
Scale X: 1 Y: 1 Z: 1

Canvas Renderer

Cull Transparent Mesh

TextMeshPro - Text (UI)

Text Input Enable RTL Editor

Text Style Normal

Main Settings

Font Asset Liberation Sans SDF (TMP\_Font Asset)  
Material Preset Liberation Sans SDF Material

Font Style B I U S ab AB SC

Font Size 0.48

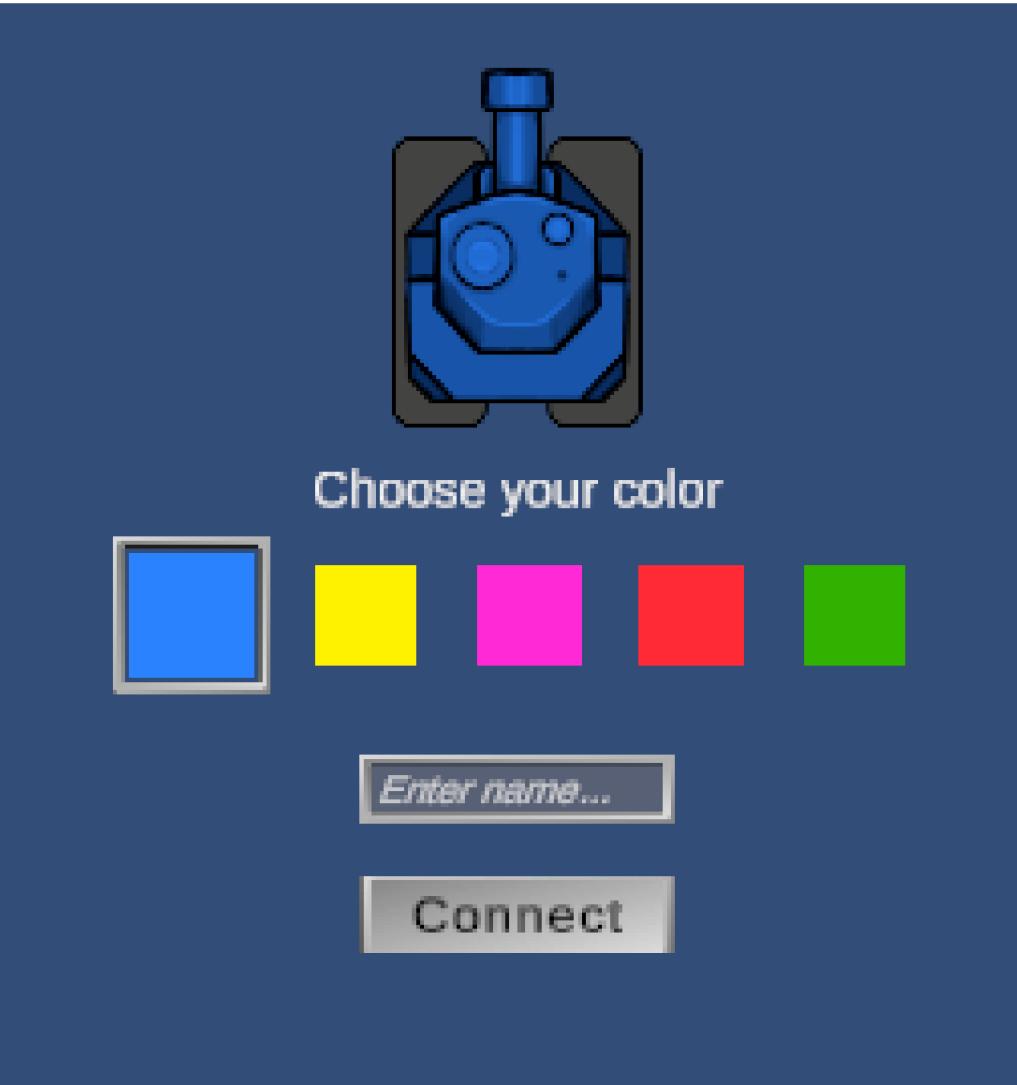
Auto Size ✓

Auto Size Options Min: 0 Max: 0.6 WD%: 0 Line: 0

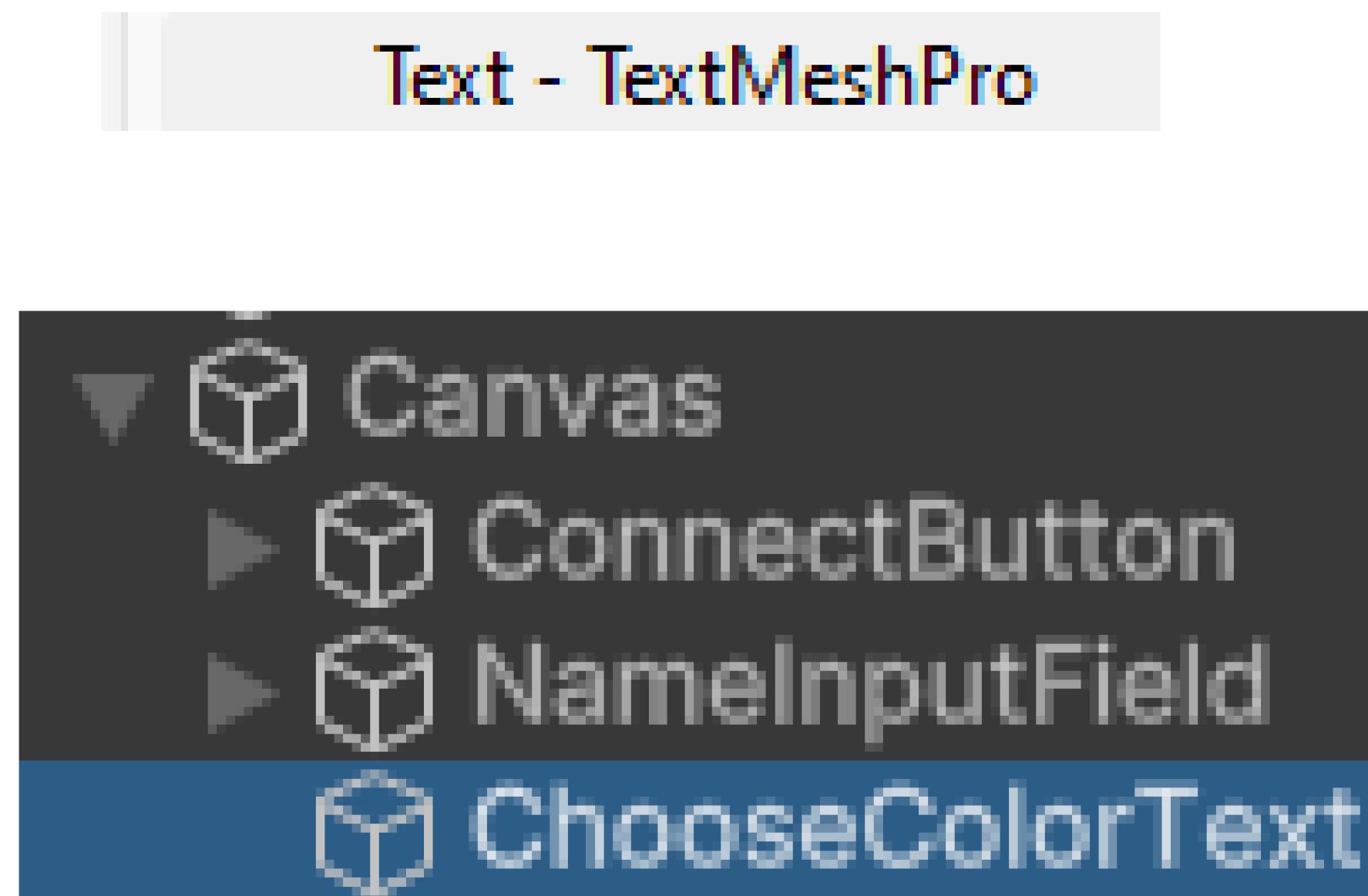
<- ถ้าเลือกเก็บไปก็สามารถนำไปรับเพิ่มได้ ตามตัวอย่างนี้



# **Change Color**



# Bootstrap



**Inspector**

**ChooseColorText**

Tag Untagged Layer Default

**Rect Transform**

center	Pos X	Pos Y	Pos Z
middle	0	230	0
	Width	Height	
	400	50	

**Anchors**

Min	X 0.5	Y 0.5
Max	X 0.5	Y 0.5
Pivot	X 0.5	Y 0.5

**Rotation** X 0 Y 0 Z 0

**Scale** X 1 Y 1 Z 1

**Canvas Renderer**

Cull Transparent Mesh

**TextMeshPro - Text (UI)**

**Text Input** Enable RTL Editor

Choose your color

**Main Settings**

Font Asset	LiberationSans SDF (TMP_Font Asset)
Material Preset	LiberationSans SDF Material
Font Style	B I U S ab AB SC
Font Size	32
Auto Size	<input type="checkbox"/>

**Vertex Color**

**Color Gradient**

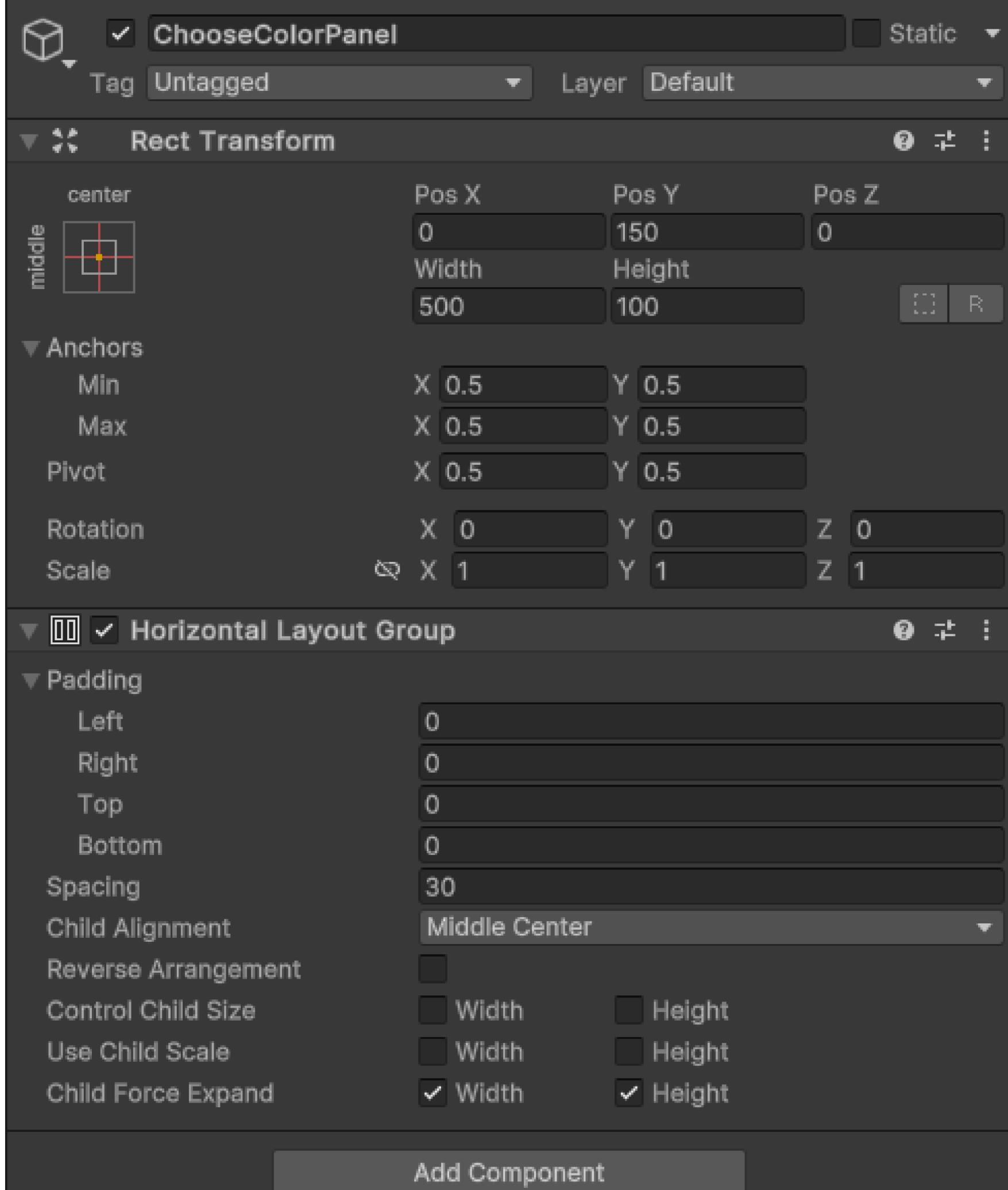
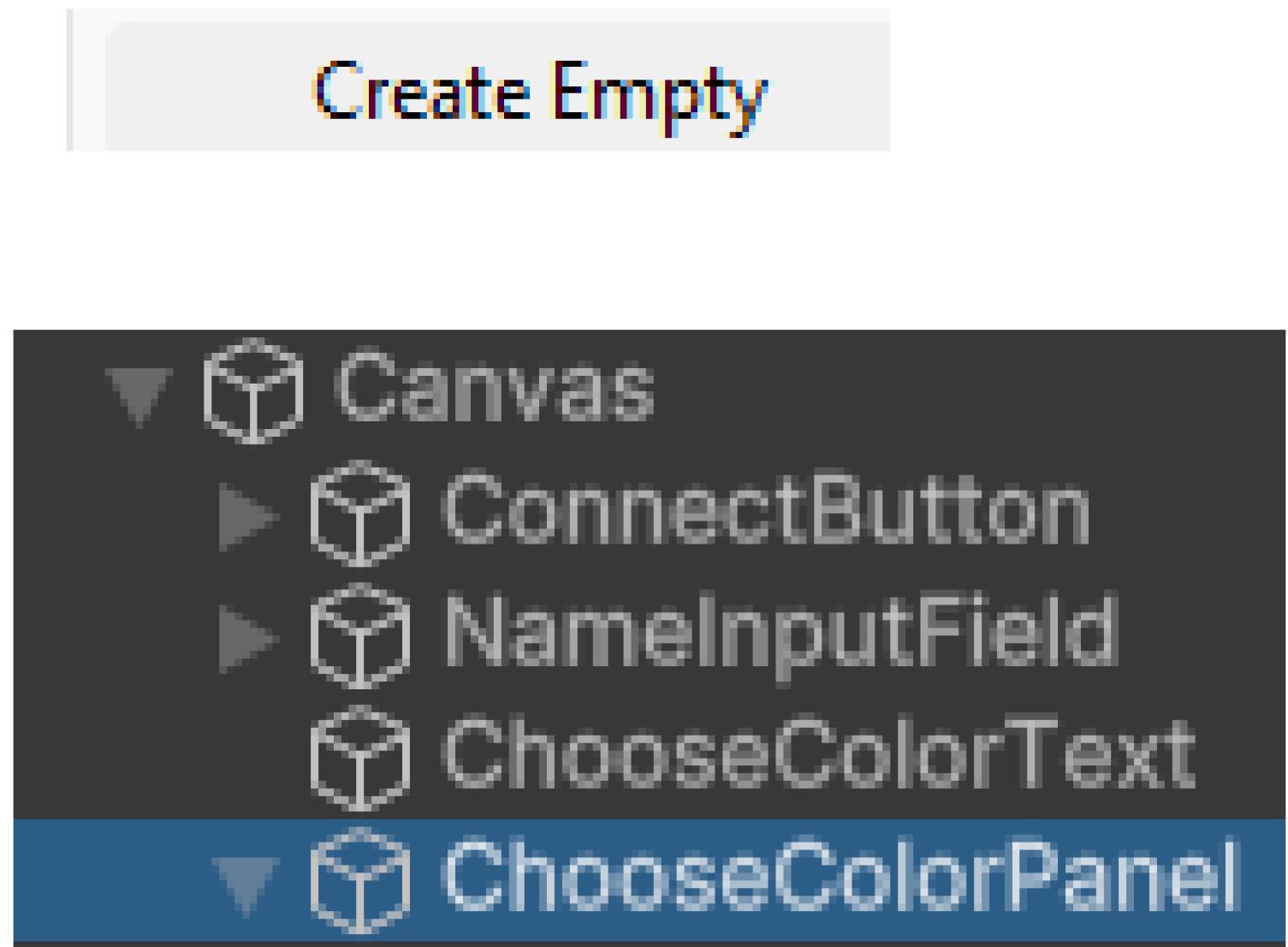
**Override Tags**

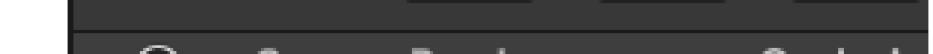
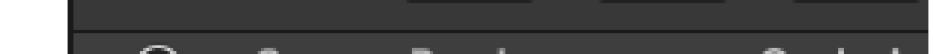
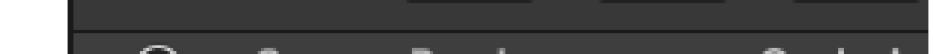
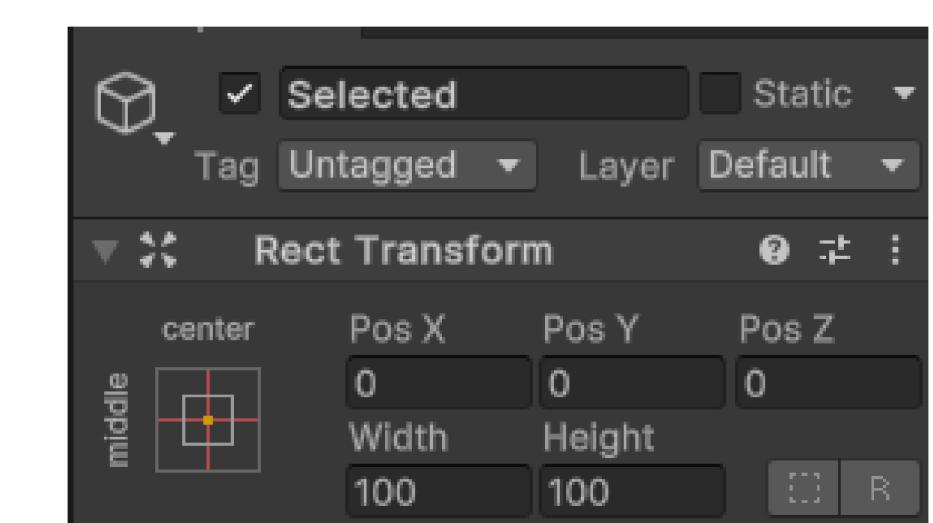
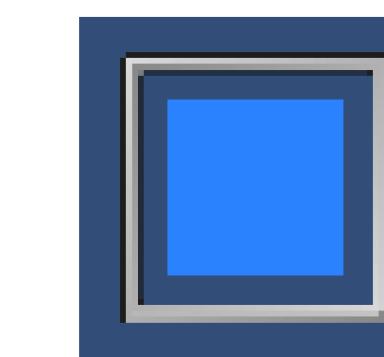
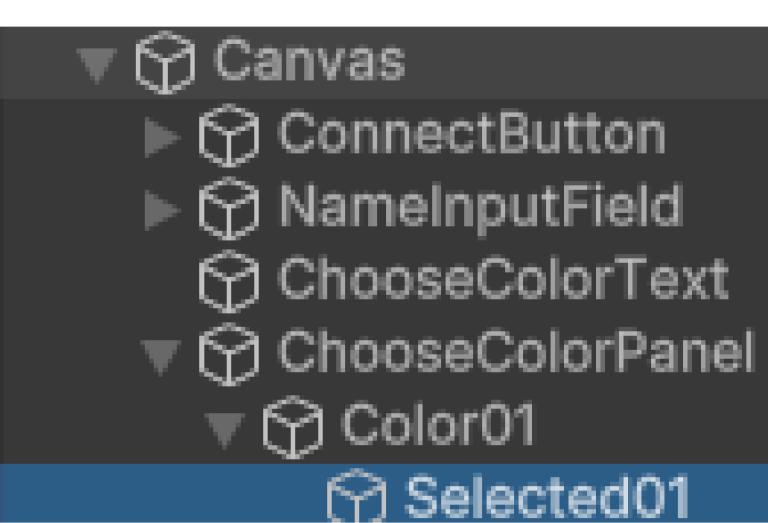
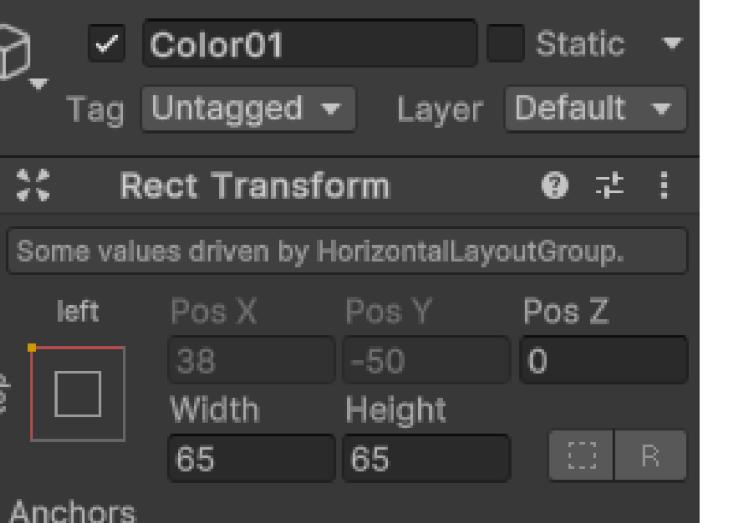
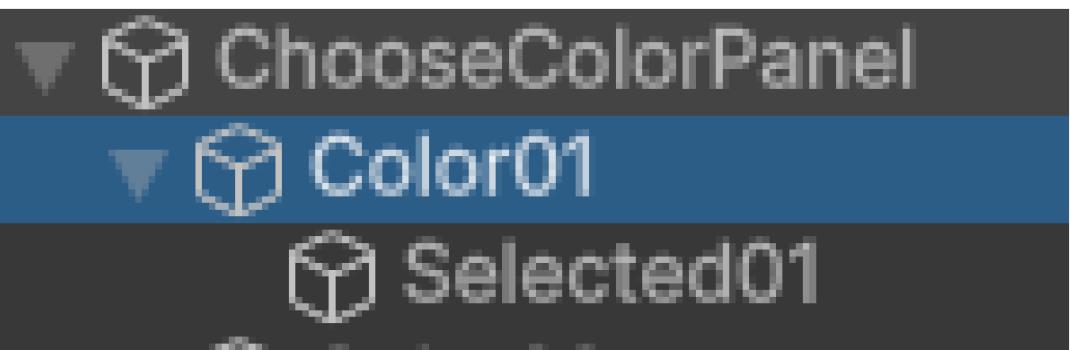
**Spacing Options (em)** Character 0 Word 0

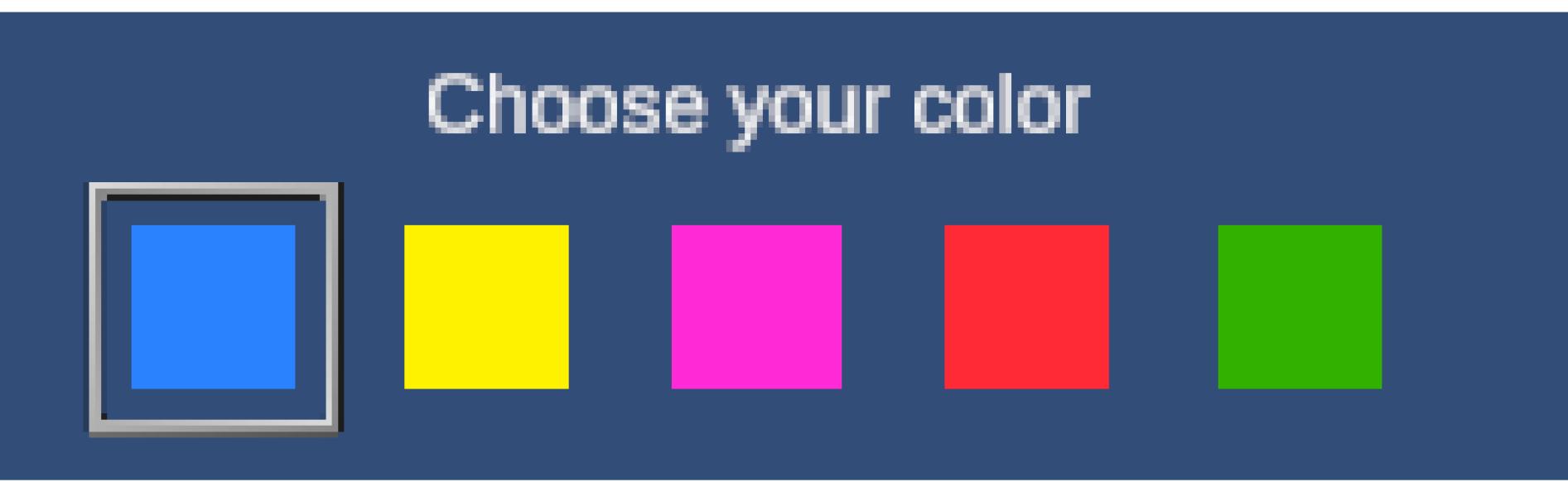
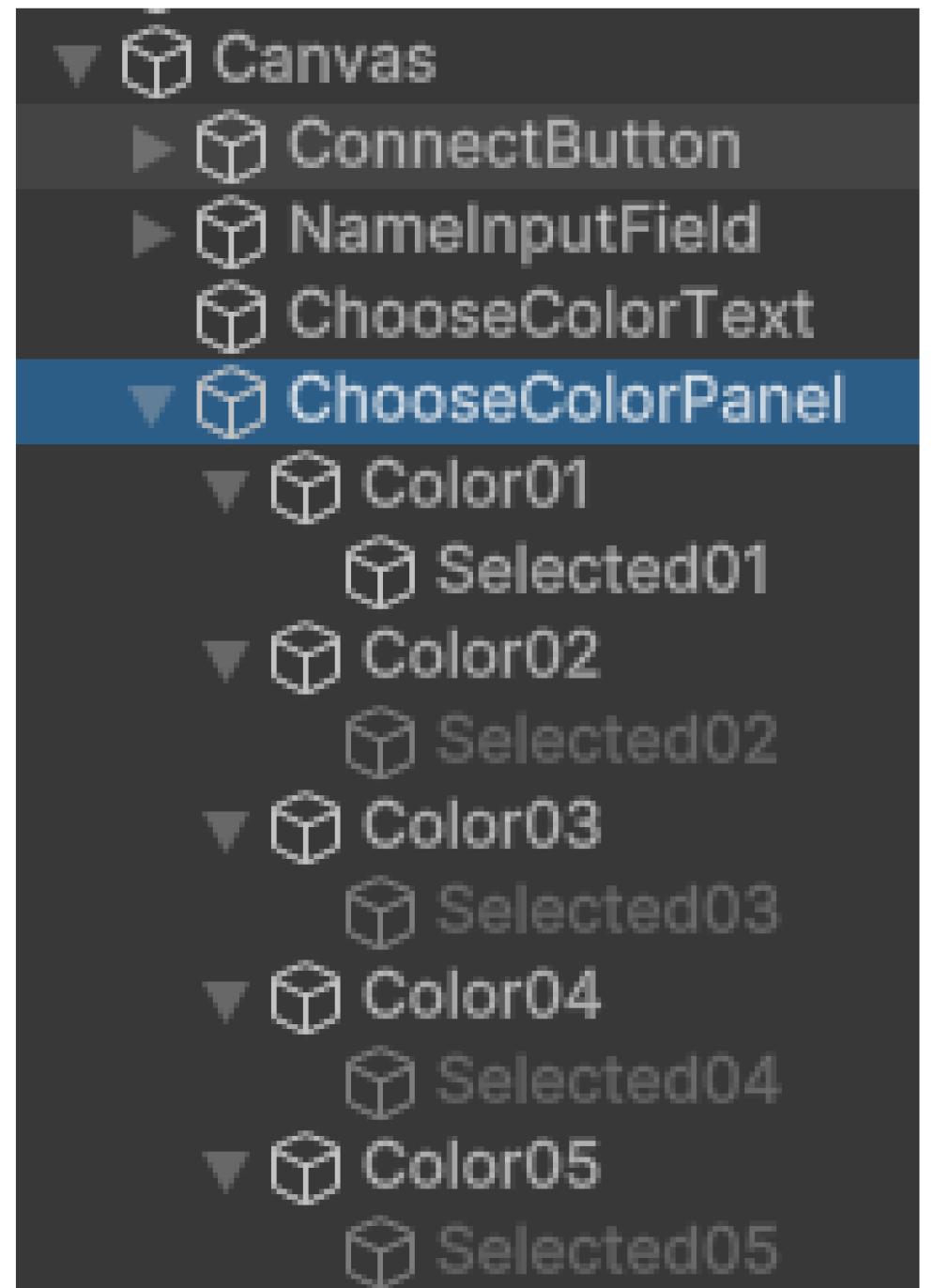
Line 0 Paragraph 0

**Alignment**

Left	Center	Right	Justify	Stretch
Top	Bottom	Top	Bottom	Top

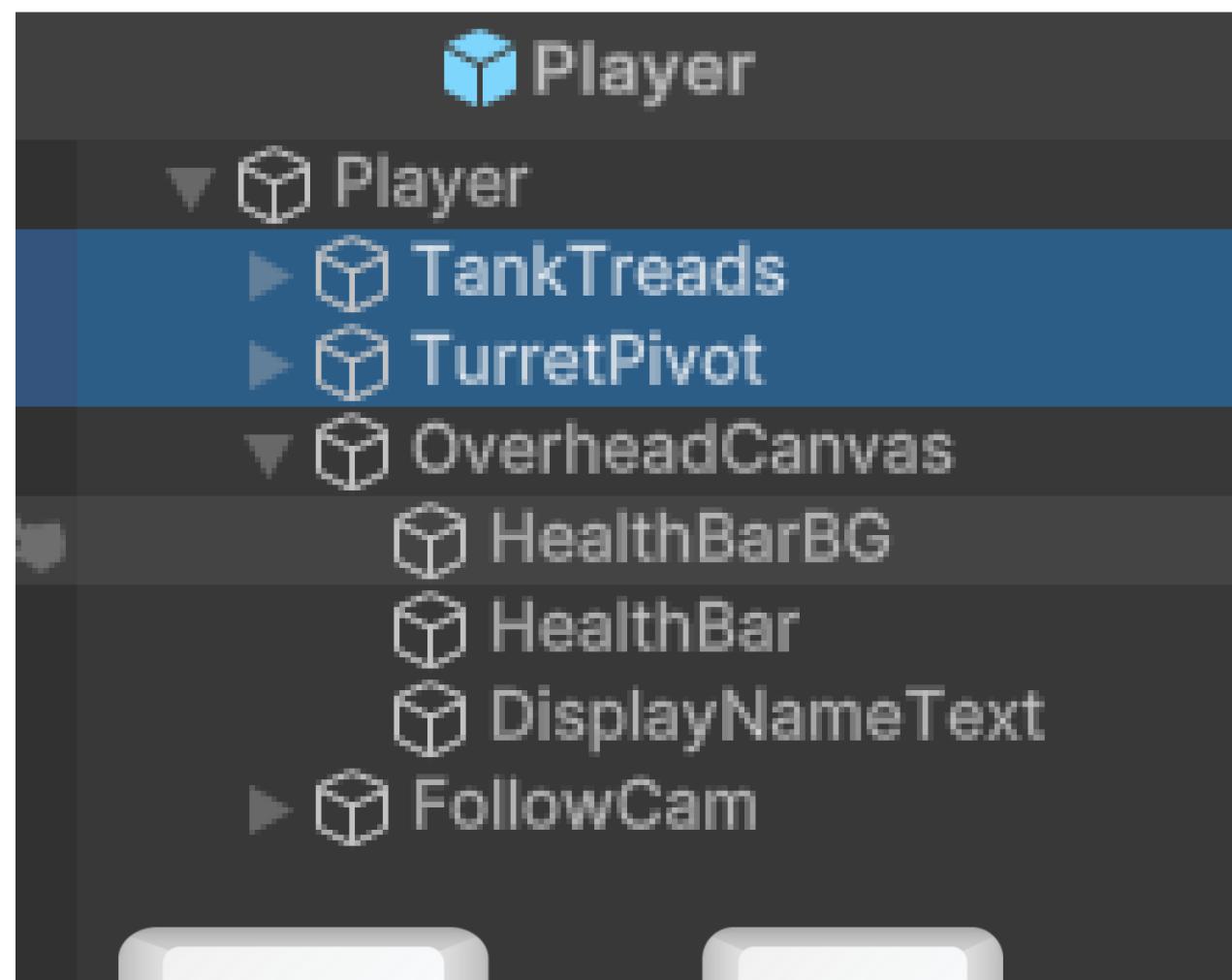
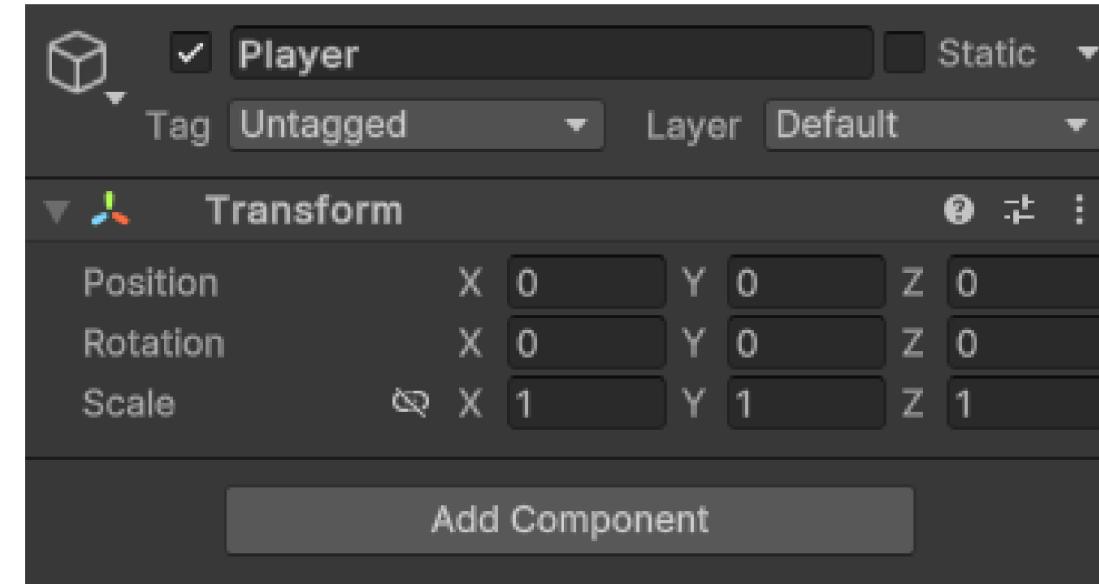






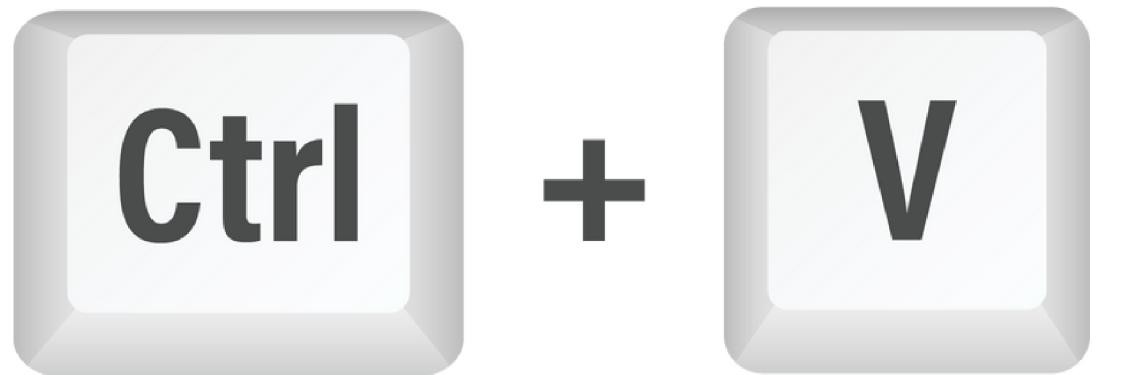
Create Empty

Player

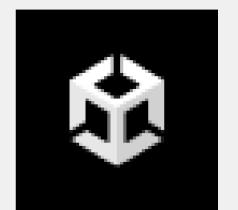


Ctrl + C

## Hierarchy



NetworkBehaviours require a NetworkObject

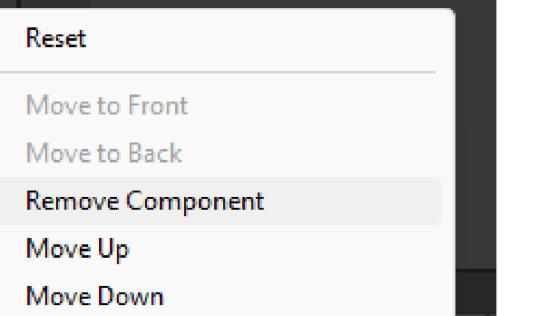
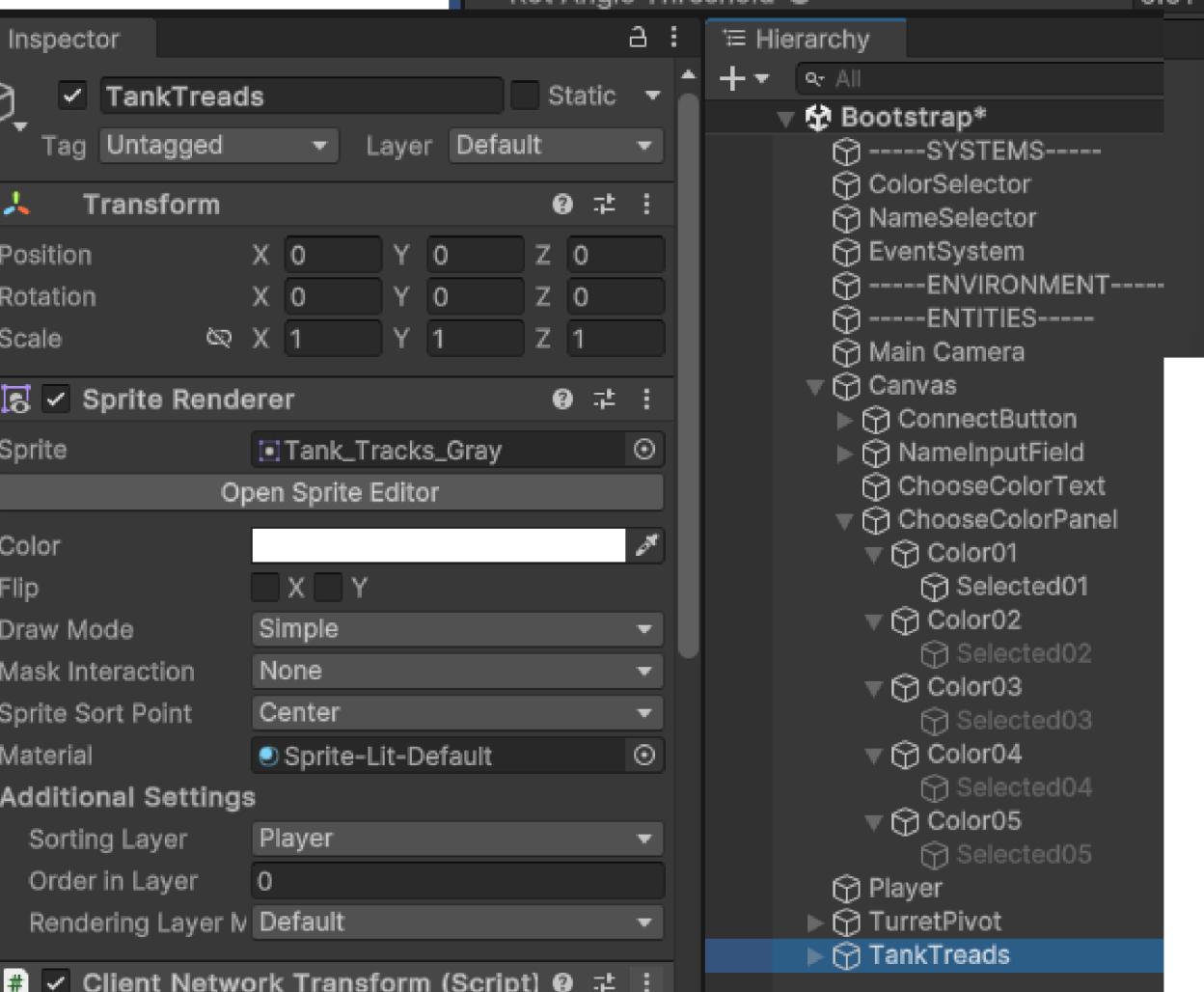
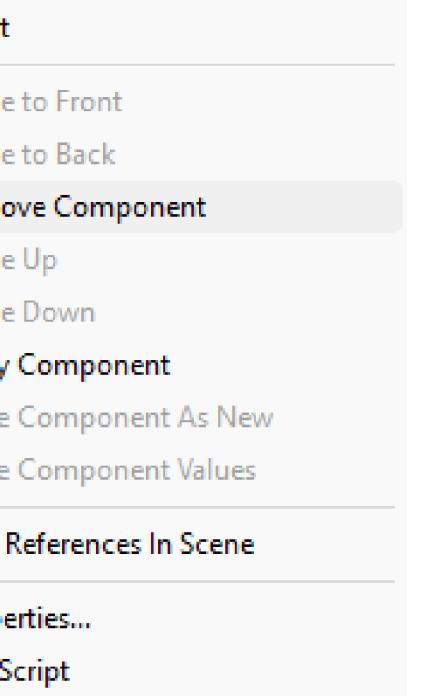
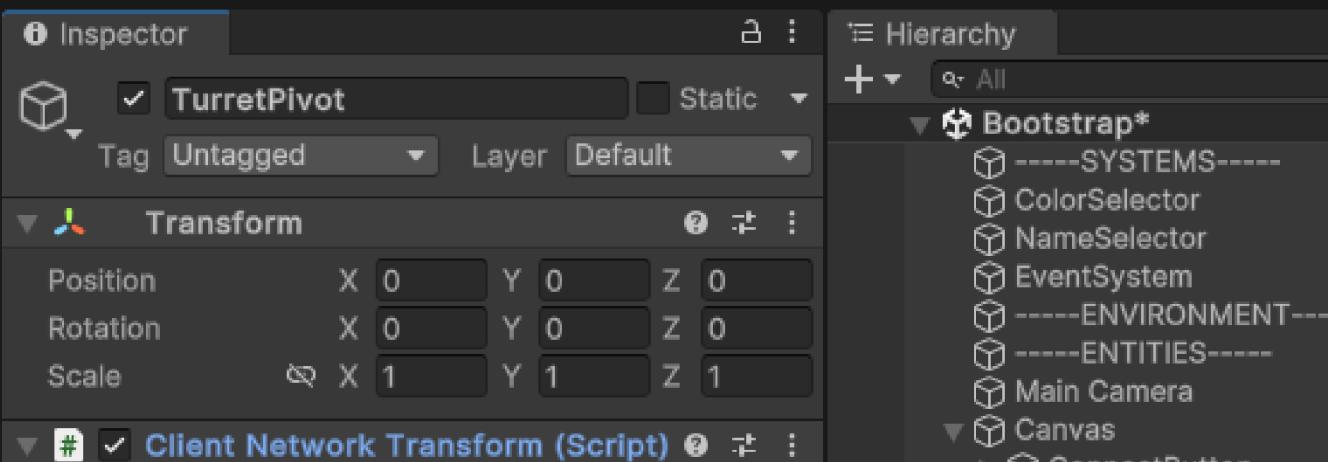


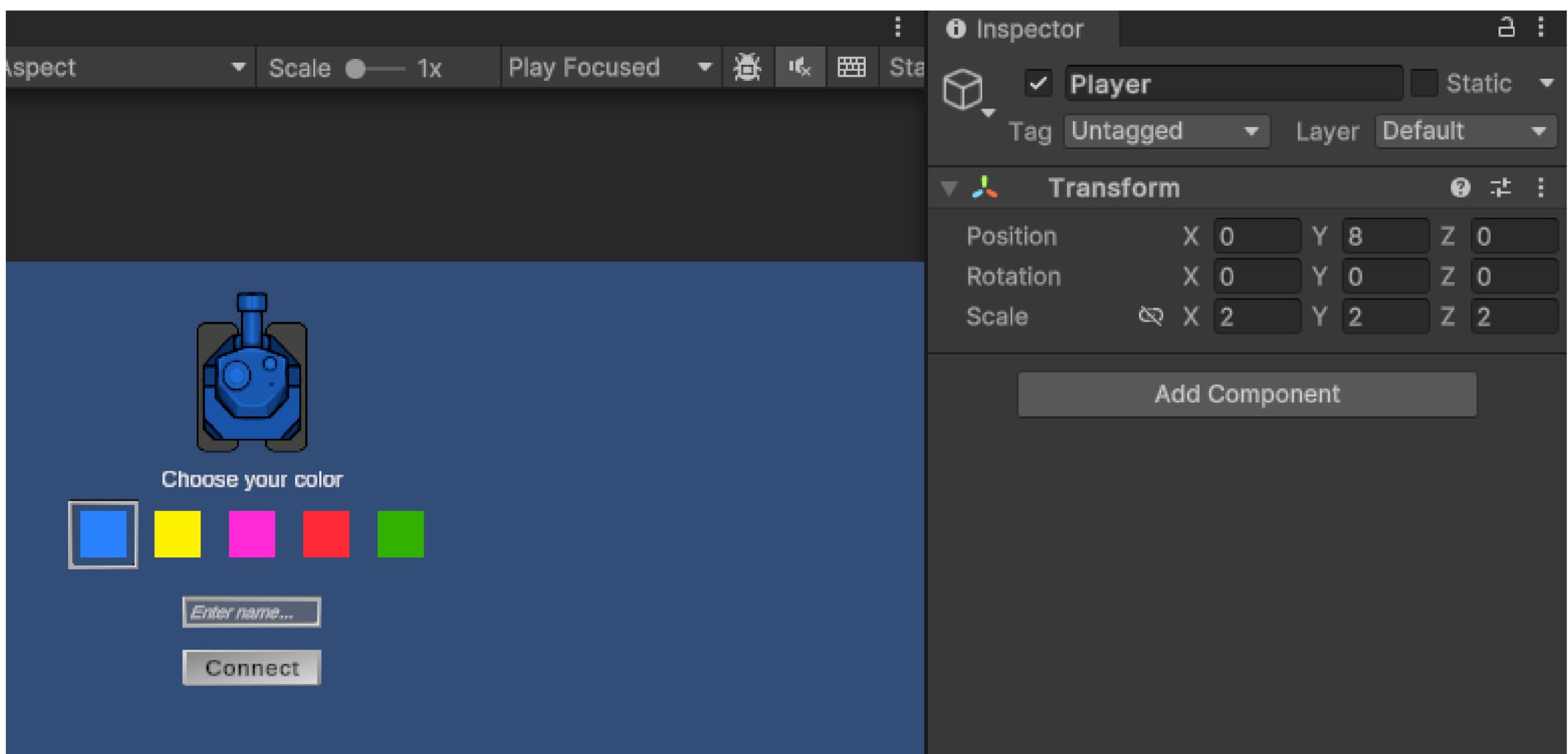
TurretPivot does not have a NetworkObject component. Would you like to add one now?

[Yes](#)

[Yes - Do not show me this message again on this machine.](#)

[No \(manually add it\)](#)





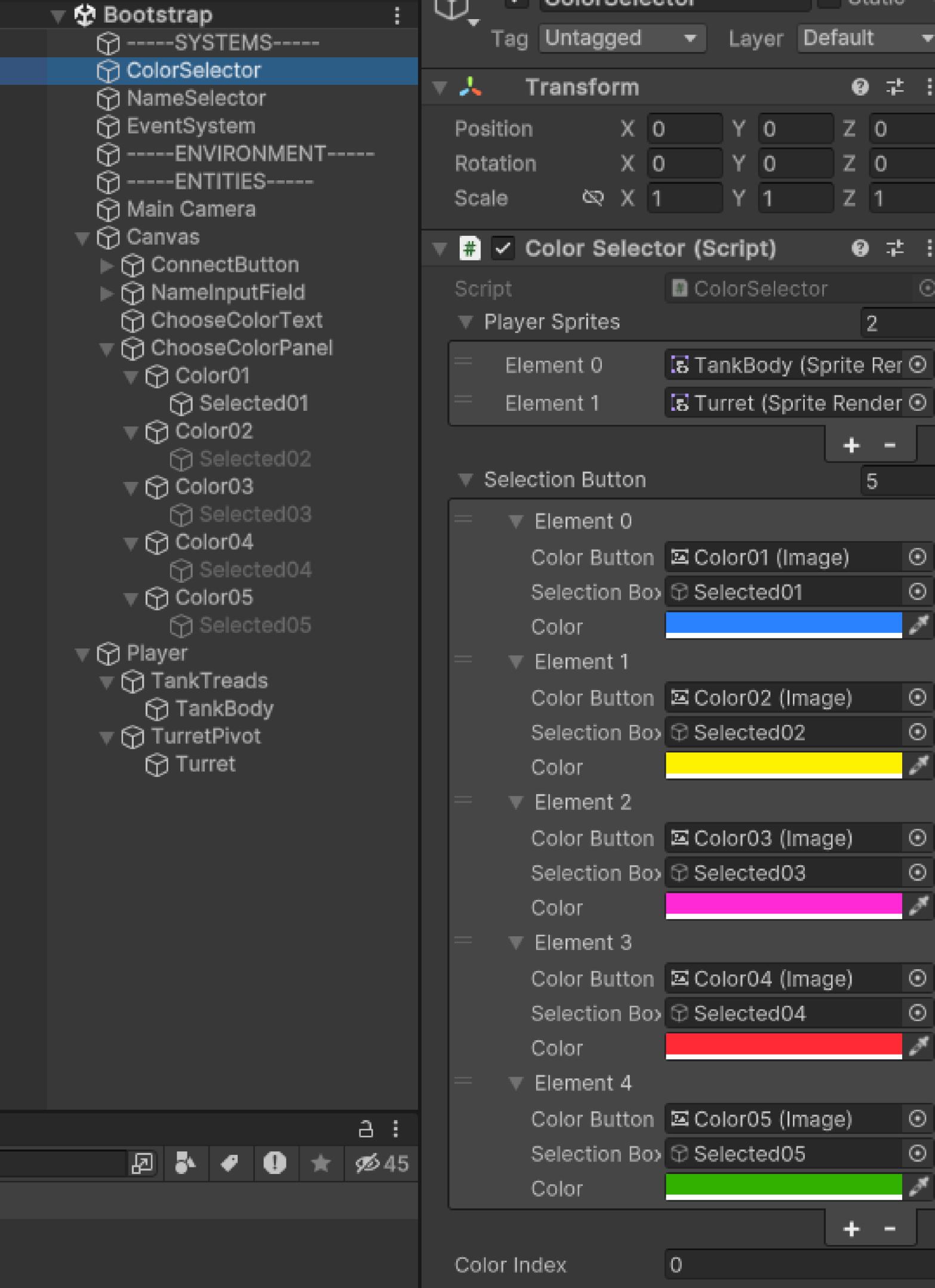


# Assets/Scripts/UI/ColorSelector.cs

## ColorSelector.cs

```
Assembly-CSharp ColorSelector OnValidate()
```

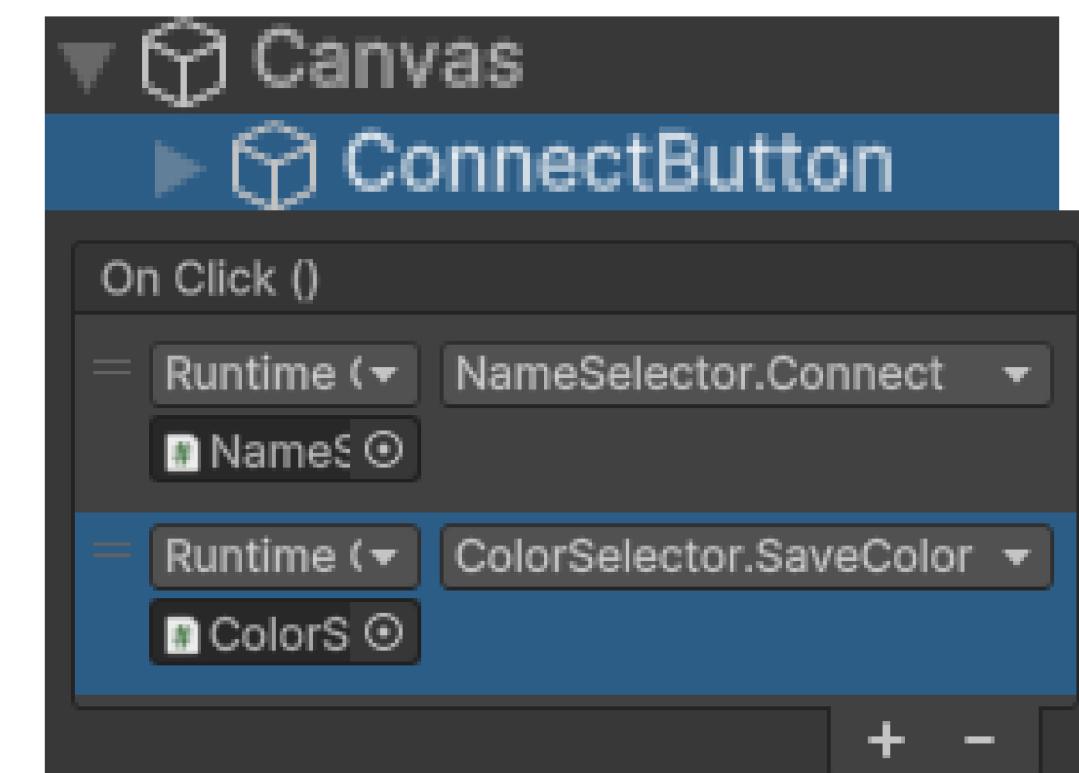
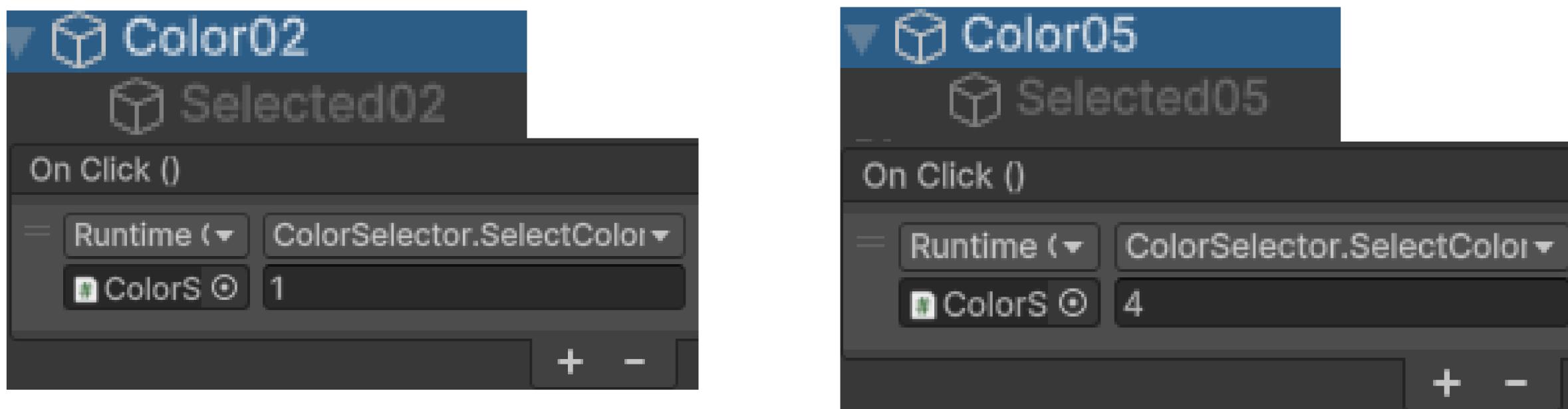
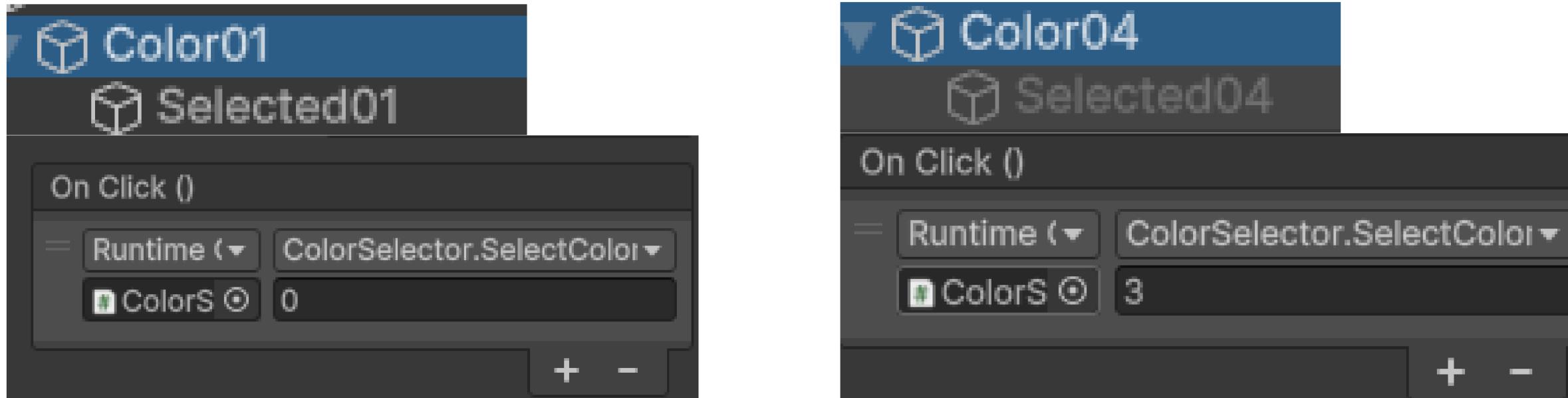
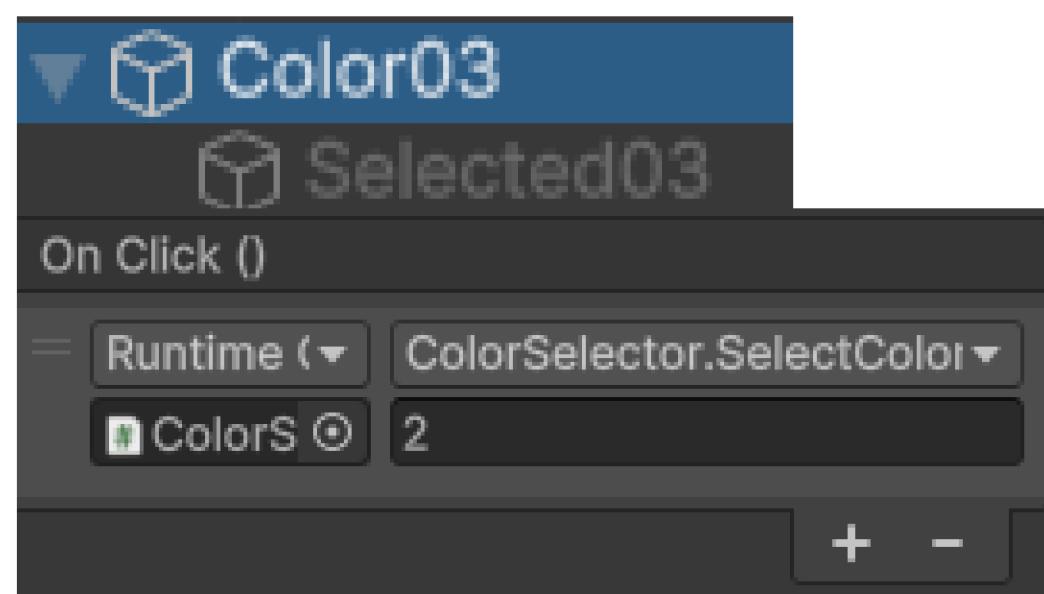
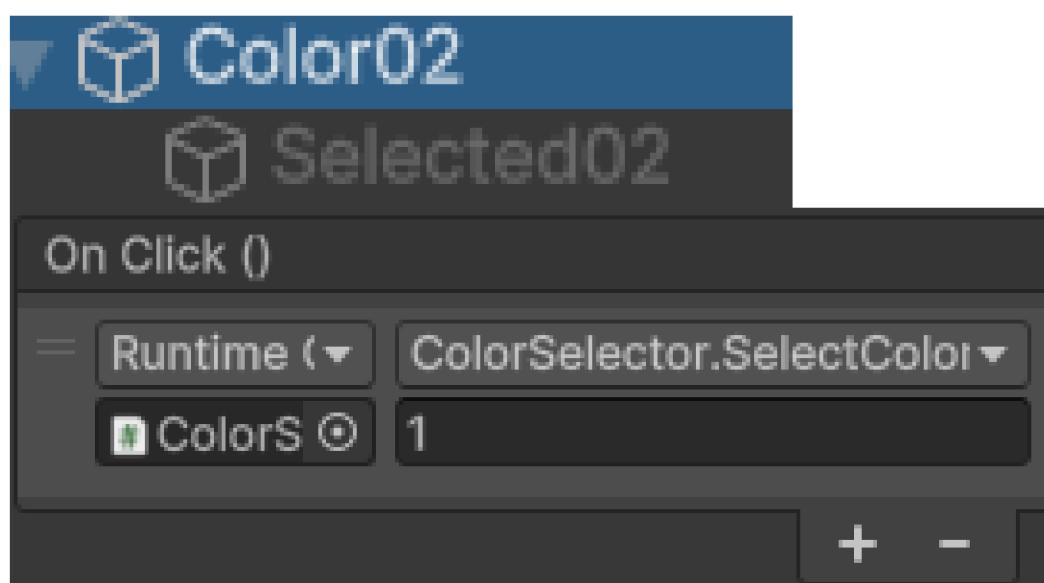
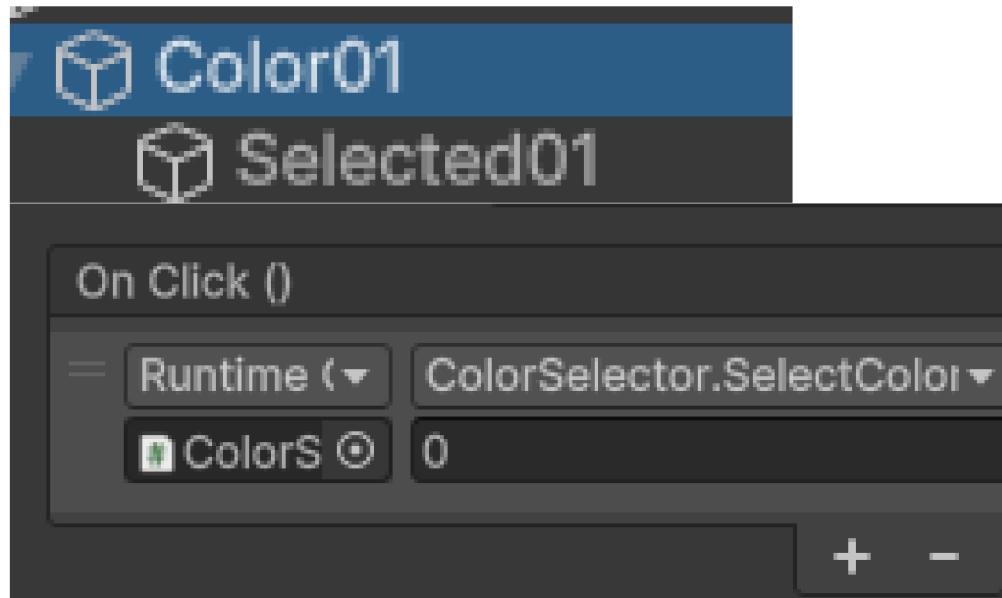
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  [System.Serializable]
7  public struct SelectionButton
8  {
9      public Image colorButton;
10     public GameObject selectionBox;
11     public Color color;
12 }
13 public class ColorSelector : MonoBehaviour
14 {
15     [SerializeField] private SpriteRenderer[] playerSprites;
16     [SerializeField] private SelectionButton[] selectionButton;
17     [SerializeField] private int colorIndex=0;
18
19     public const string PlayerColorKey = "PlayerColorIndex";
20 }
```



## ColorSelector.cs

```
private void OnValidate()
{
    foreach (var selection in selectionButton)
    {
        selection.colorButton.color = selection.color;
    }
}
```

```
private void Start()
{
    colorIndex = PlayerPrefs.GetInt(PlayerColorKey, 0);
    HandleColorChanged();
}
2 references
public void HandleColorChanged()
{
    foreach (var selection in selectionButton)
    {
        selection.selectionBox.SetActive(false);
    }
    foreach (var sprite in playerSprites)
    {
        sprite.color = selectionButton[colorIndex].color;
    }
    selectionButton[colorIndex].selectionBox.SetActive(true);
}
0 references
public void SelectColor(int colorIndex)
{
    this.colorIndex = colorIndex;
    HandleColorChanged();
}
0 references
public void SaveColor()
{
    PlayerPrefs.SetInt(PlayerColorKey, colorIndex);
}
```



```
1  using System;
2
3  [Serializable]
4  public class UserData
5  {
6      public string userName;
7      public string userAuthId;
8      public int userColorIndex;
9  }
```

เมื่อมีการเพิ่ม Data ใหม่เข้ามา จำดับต่อไป  
จำเป็นต้อง Setup ค่าเข้าไปใน UserData กี่ถูก new ด้วย

“ C# ClientGameManager.cs & C# HostGameManager.cs ”

```
UserData userData = new UserData
{
    userName = PlayerPrefs.GetString(NameSelector.PlayerNameKey, "Missing Name"),
    userAuthId = AuthenticationService.Instance.PlayerId,
    userColorIndex = PlayerPrefs.GetInt(ColorSelector.PlayerColorKey, 0)
};
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Unity.Netcode;
4  using UnityEngine;
5  using Cinemachine;
6  using Unity.Collections;
7
8  ⚡ Unity Script (1 asset reference) | 1 reference
9  public class TankPlayer : NetworkBehaviour
10 {
11     [Header("References")]
12     [SerializeField] private CinemachineVirtualCamera virtualCamera;
13
14     [Header("Settings")]
15     [SerializeField] private int ownerPriority = 15;
16
17     public NetworkVariable<FixedString32Bytes> PlayerName = new NetworkVariable<FixedString32Bytes>();
18     public NetworkVariable<int> PlayerColorIndex = new NetworkVariable<int>();
19     0 references
20
21     public override void OnNetworkSpawn()
22     {
23         if (IsServer)
24         {
25             UserData userData =
26                 HostSingleton.Instance.GameManager.NetworkServer.GetUserDataByClientId(OwnerId);
27
28             PlayerName.Value = userData.userName;
29             PlayerColorIndex.Value = userData.userColorIndex;
30         }
31
32         if (IsOwner)
33         {
34             virtualCamera.Priority = ownerPriority;
35         }
36     }
37 }
```



# Assets/Scripts/Core/Player/PlayerColor.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerColor : MonoBehaviour
6  {
7      [SerializeField] private TankPlayer player;
8      [SerializeField] private SpriteRenderer[] playerSprites;
9      [SerializeField] private Color[] tankColor;
10     [SerializeField] private int colorIndex;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         HandlePlayerColorChanged(0, player.PlayerColorIndex.Value);
16         player.PlayerColorIndex.OnValueChanged += HandlePlayerColorChanged;
17     }
18
19     private void HandlePlayerColorChanged(int oldIndex, int newIndex)
20     {
21         colorIndex = newIndex;
22         foreach (var sprite in playerSprites)
23         {
24             sprite.color = tankColor[colorIndex];
25         }
26     }
27
28     private void OnDestroy()
29     {
30         player.PlayerColorIndex.OnValueChanged -= HandlePlayerColorChanged;
31     }
}
```

Hierarchy

All

Player

- Player
- TankTreads
- TankBody
- TurretPivot
- Turret
  - ProjectileSpawnPoint
  - MuzzleFlash
- OverheadCanvas
  - HealthBarBG
  - HealthBar
  - DisplayNameText
- FollowCam

Inspector

# Player Color (Script)

Script: PlayerColor

Player: Player (Tank PI)

Player Sprites: 2

- Element 0: TankBody (Sprite Renderer)
- Element 1: Turret (Sprite Renderer)

+ -

Tank Color: 5

- Element 0: Blue
- Element 1: Yellow
- Element 2: Magenta
- Element 3: Red
- Element 4: Green

+ -

Color Index: 0

45

Player

Inspector

ColorSelector

Static: Unchecked

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

# Color Selector (Script)

Script: ColorSelector

Player Sprites: 2

- Element 0: TankBody (Sprite Renderer)
- Element 1: Turret (Sprite Renderer)

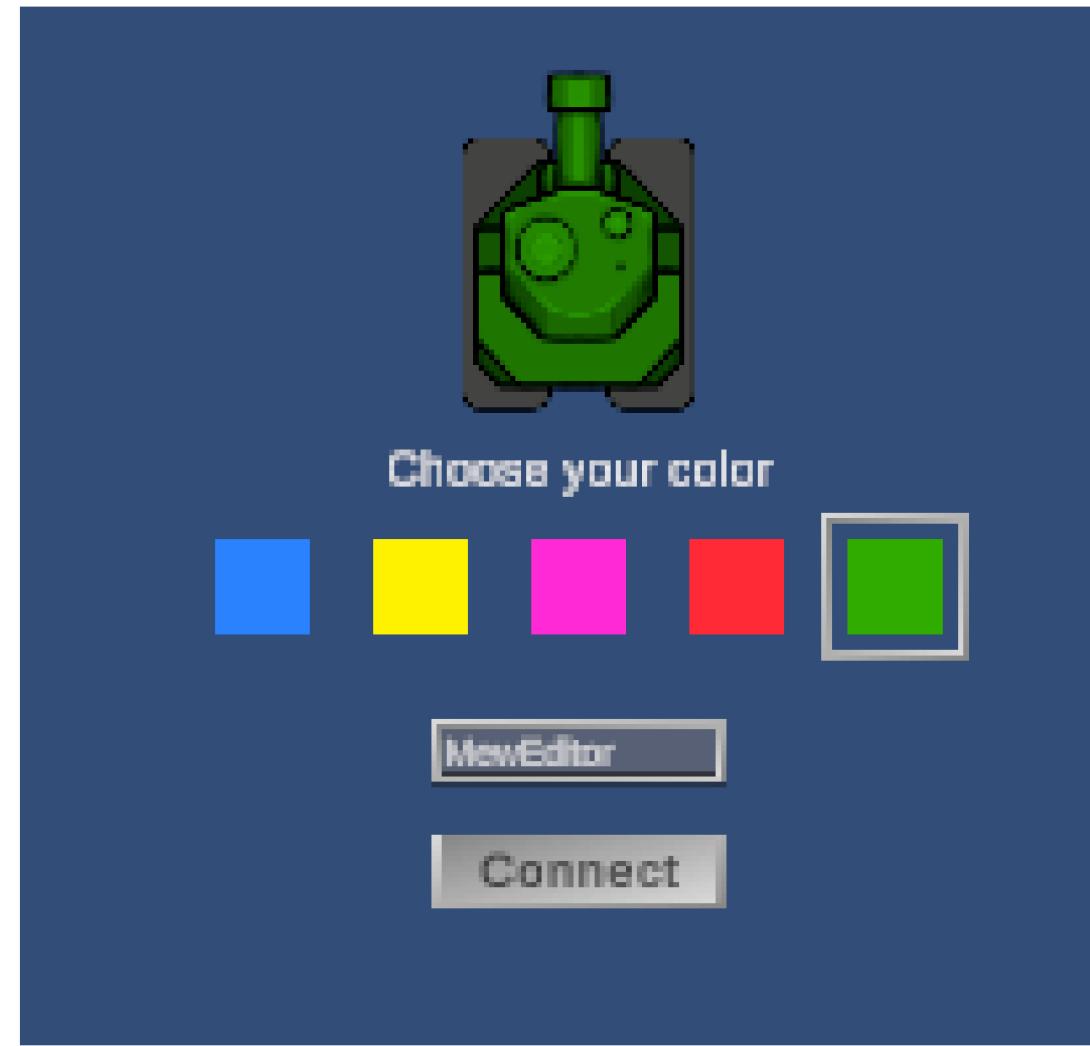
+ -

Selection Button: 5

- Element 0
  - Color Button: Color01 (Image)
  - Selection Box: Selected01
  - Color: Blue
- Element 1
  - Color Button: Color02 (Image)
  - Selection Box: Selected02
  - Color: Yellow
- Element 2
  - Color Button: Color03 (Image)
  - Selection Box: Selected03
  - Color: Magenta
- Element 3
  - Color Button: Color04 (Image)
  - Selection Box: Selected04
  - Color: Red
- Element 4
  - Color Button: Color05 (Image)
  - Selection Box: Selected05
  - Color: Green

+ -

Color Index: 0

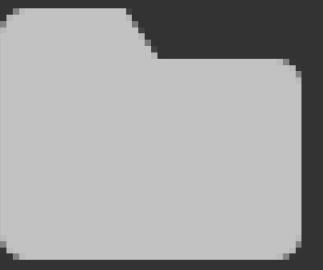


# **Spawn Points**

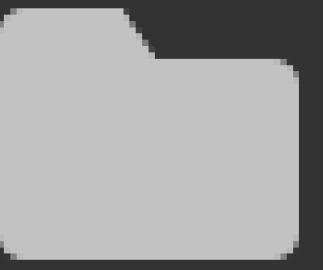
Assets > Scripts > Core



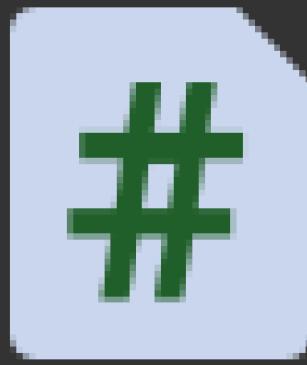
Coins



Combat



Player

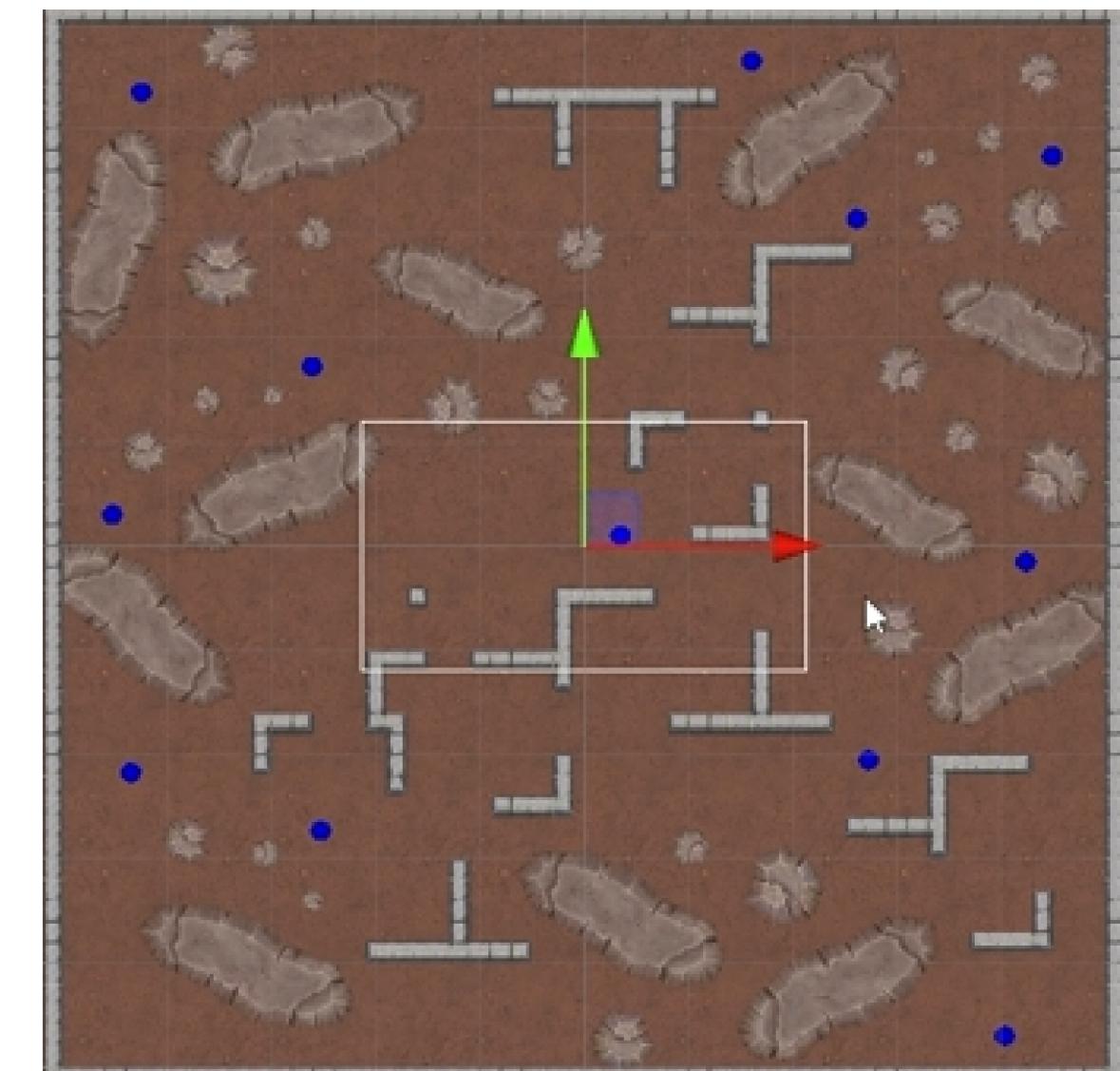


SpawnPoint

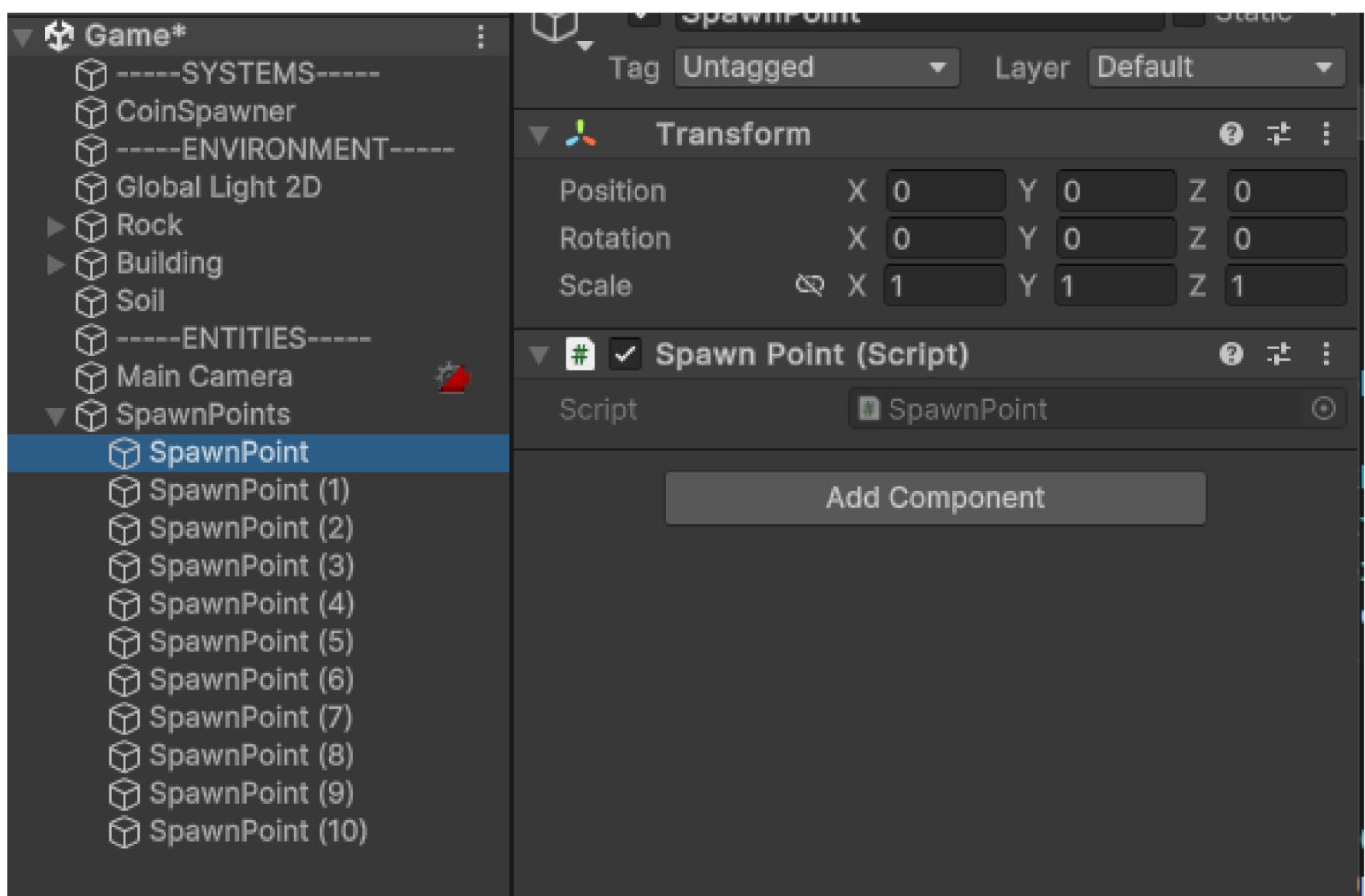
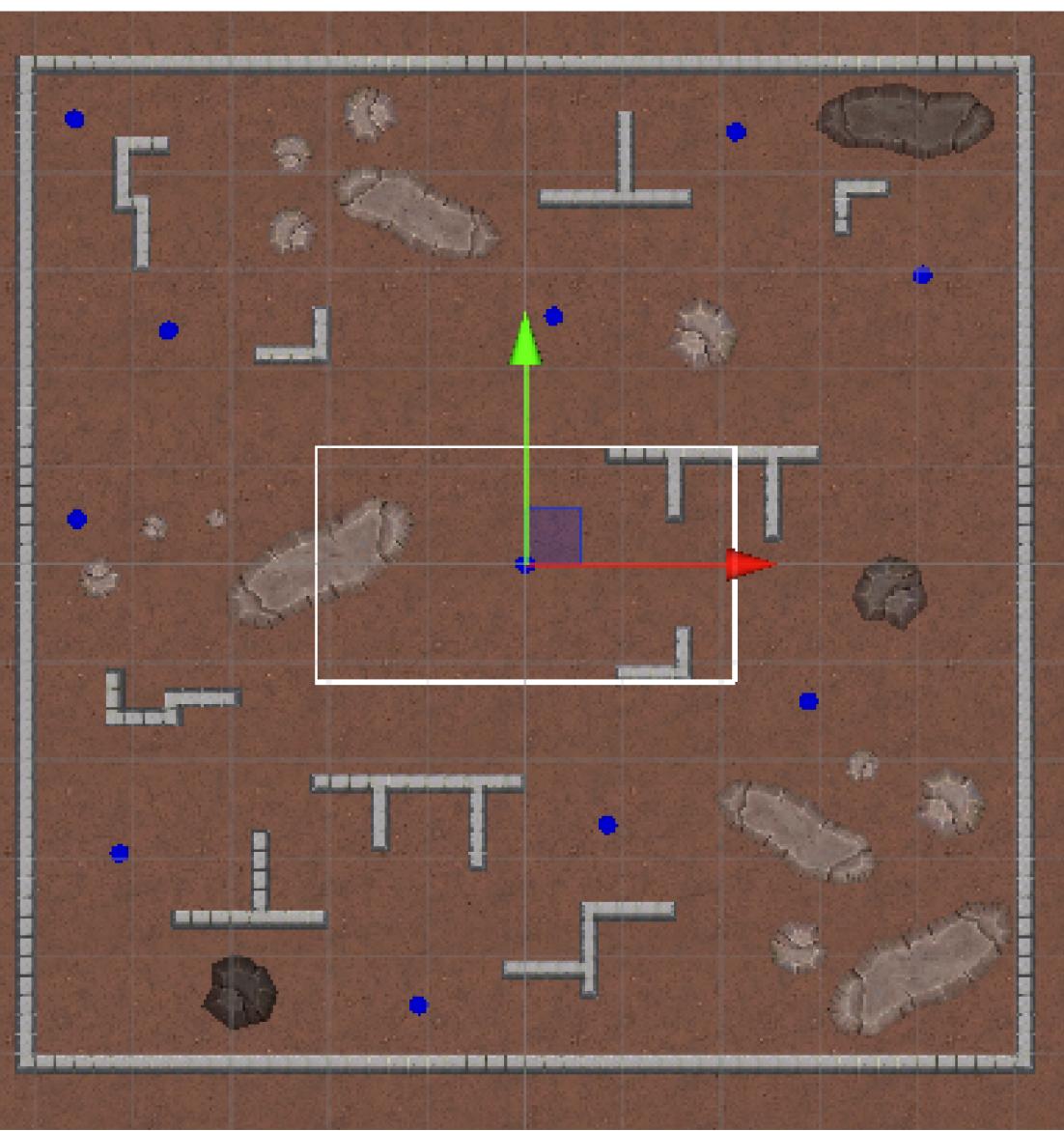
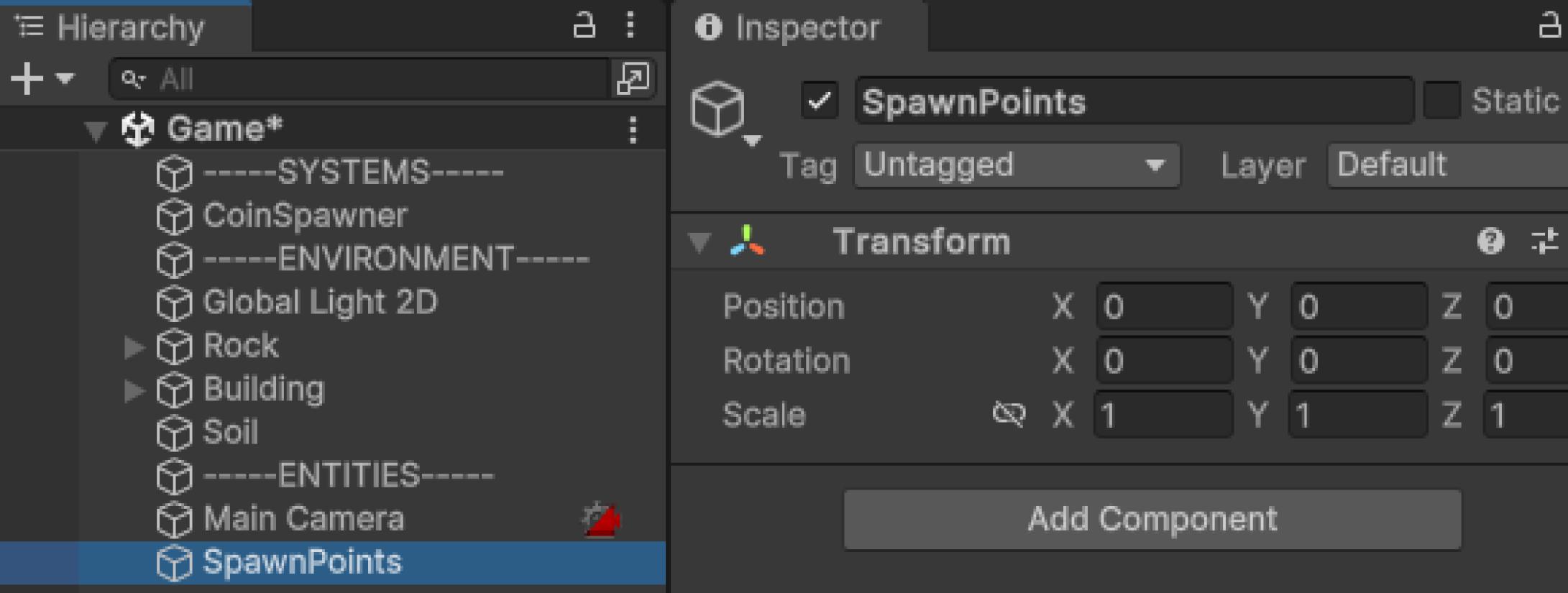
```
1  [-]using System.Collections;
2  [-]using System.Collections.Generic;
3  [-]using UnityEngine;
4
5  [-]④ Unity Script | 2 references
6  [-]public class SpawnPoint : MonoBehaviour
7  [-]{ 
8
9  [-]    [-]private static List<SpawnPoint> spawnPoints = new List<SpawnPoint>();
10
11
12
13  [-]}
```

# Spawn Point

- Add **this** spawn point to the list in `OnEnable`
- Remove **this** spawn point from the list in `OnDisable`
- In the **GetRandomSpawnPos** method, return a random spawn point's position from the list



```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  #region Unity Script | 2 references
6  public class SpawnPoint : MonoBehaviour
7  {
8
9      #region Unity Message | 0 references
10     private void OnEnable()
11     {
12         spawnPoints.Add(this);
13     }
14
15     #region Unity Message | 0 references
16     private void OnDisable()
17     {
18         spawnPoints.Remove(this);
19     }
20
21     #endregion Unity Message | 0 references
22
23     public static Vector3 GetRandomSpawnPos()
24     {
25
26         #region Unity Message | 0 references
27         if(spawnPoints.Count == 0)
28         {
29             return Vector3.zero;
30         }
31
32         return spawnPoints[Random.Range(0, spawnPoints.Count)].transform.position;
33     }
34
35     #endregion Unity Message | 0 references
36
37     private void OnDrawGizmos()
38     {
39         Gizmos.color = Color.blue;
40         Gizmos.DrawSphere(transform.position, 1);
41     }
42
43 }
```



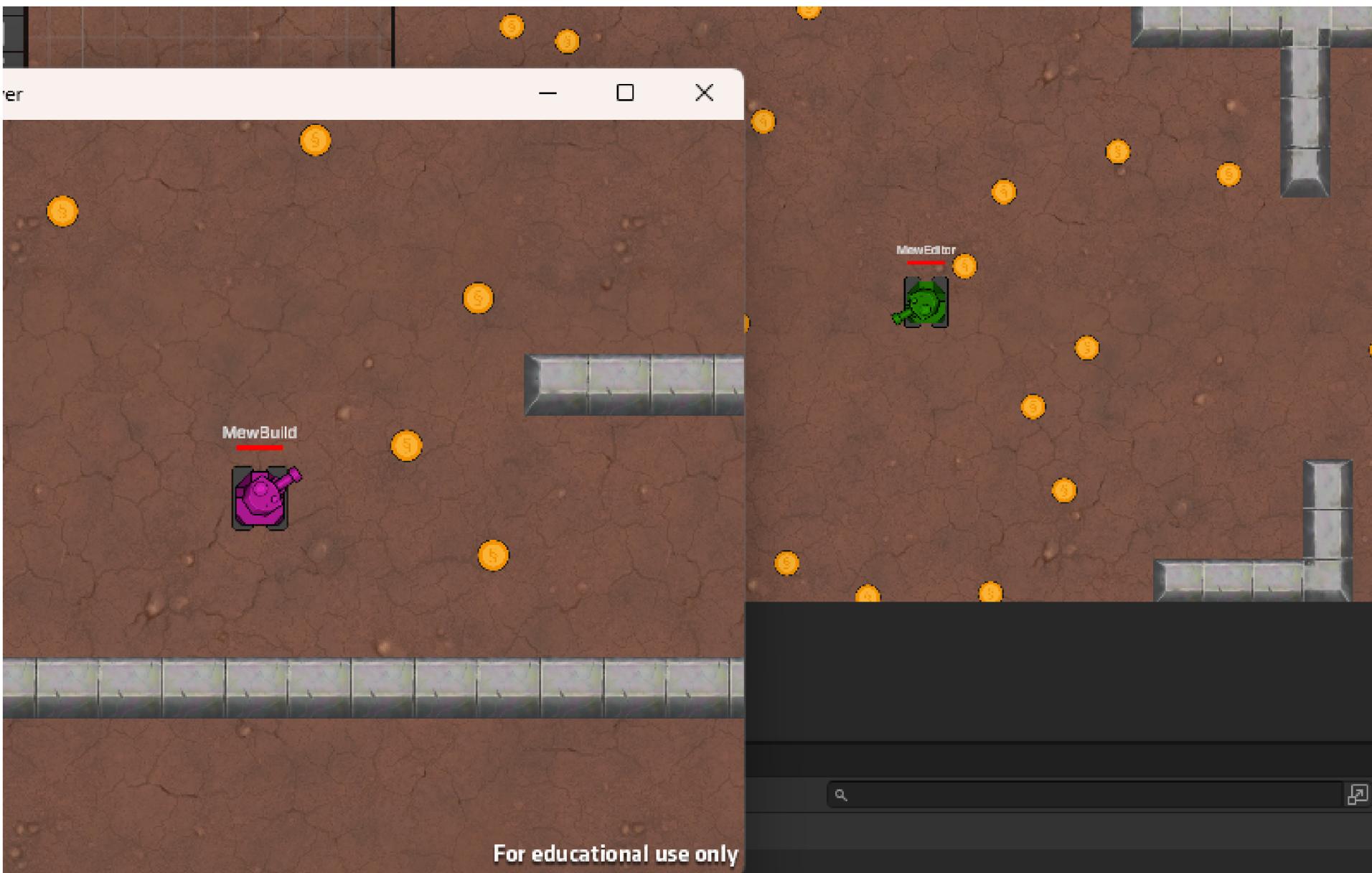


# NetworkServer.cs

```
2 references
private void ApprovalCheck(
    NetworkManager.ConnectionApprovalRequest request,
    NetworkManager.ConnectionApprovalResponse response)
{
    string payload = System.Text.Encoding.UTF8.GetString(request.Payload);
    UserData userData = JsonUtility.FromJson<UserData>(payload);

    clientIdToAuth[request.ClientNetworkId] = userData.userAuthId;
    authIdToUserData(userData.userAuthId) = userData;
    //Debug.Log(userData.userName);

    response.Approved = true;
    response.Position = SpawnPoint.GetRandomSpawnPos();
    response.Rotation = Quaternion.identity;
    response.CreatePlayerObject = true;
}
```



For educational use only

# **Respawning**

```
Unity Script (1 asset reference) | 4 references
9  public class TankPlayer : NetworkBehaviour
10 {
11     [Header("References")]
12     [SerializeField] private CinemachineVirtualCamera virtualCamera;
13
14     [Header("Settings")]
15     [SerializeField] private int ownerPriority = 15;
16
17     public NetworkVariable<FixedString32Bytes> PlayerName = new NetworkVariable<FixedString32Bytes>();
18     public NetworkVariable<int> PlayerColorIndex = new NetworkVariable<int>();
19
20     public static event Action<TankPlayer> OnPlayerSpawned;
21     public static event Action<TankPlayer> OnPlayerDespawned;

public override void OnNetworkSpawn()
{
    if (IsServer)
    {
        UserData userData =
            HostSingleton.Instance.GameManager.NetworkServer.GetUserDataByClientId(OwnerId);

        PlayerName.Value = userData.userName;
        PlayerColorIndex.Value = userData.userColorIndex;

        OnPlayerSpawned?.Invoke(this);
    }

    if (IsOwner)
    {
        virtualCamera.Priority = ownerPriority;
    }
}

0 references
public override void OnNetworkDespawn()
{
    if(IsServer)
    {
        OnPlayerDespawned?.Invoke(this);
    }
}
```

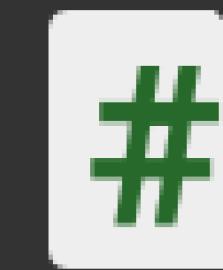
Assets > Scripts > Core > Combat



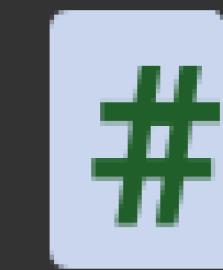
DealDamage...



Health



HealthDisp...



RespawnH...

# Assets/Scripts/Core/Combat/RespawnHandler.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Unity.Netcode;
4  using UnityEngine;
5
6  public class RespawnHandler : NetworkBehaviour
7  {
8      [SerializeField] private NetworkObject playerPrefab;
9
10     public override void OnNetworkSpawn()
11     {
12         if (!IsServer) { return; }
13     }
14
15     public override void OnNetworkDespawn()
16     {
17         if (!IsServer) { return; }
18     }
19 }
```

```
Unity Script | 0 references
7 public class RespawnHandler : NetworkBehaviour
8 {
9     [SerializeField] private NetworkObject playerPrefab;
10
11    0 references
12    public override void OnNetworkSpawn()
13    {
14        if (!IsServer) { return; }
15        TankPlayer[] players = FindObjectsOfType<TankPlayer>(FindObjectsSortMode.None);
16        foreach (TankPlayer player in players)
17        {
18            HandlePlayerSpawned(player);
19        }
20        TankPlayer.OnPlayerSpawned += HandlePlayerSpawned;
21        TankPlayer.OnPlayerDespawned += HandlePlayerDespawned;
22    }
23
24    0 references
25    public override void OnNetworkDespawn()
26    {
27        if (!IsServer) { return; }
28        TankPlayer.OnPlayerSpawned -= HandlePlayerSpawned;
29        TankPlayer.OnPlayerDespawned -= HandlePlayerDespawned;
30    }
31    3 references
32    private void HandlePlayerSpawned(TankPlayer player)
33    {
34    }
35    2 references
36    private void HandlePlayerDespawned(TankPlayer player)
37    {
38    }
39 }
```

Unity Script (1 asset reference) | 13 references

```
public class TankPlayer : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private CinemachineVirtualCamera virtualCamera;
    0 references
    | [field:SerializeField] public Health Health { get; private set; }

    [Header("Settings")]
    [SerializeField] private int ownerPriority = 15;
```

```
private void HandlePlayerSpawned(TankPlayer player)
{
    player.Health.OnDie += (health) => HandlePlayerDie(player);
}

2 references
private void HandlePlayerDespawned(TankPlayer player)
{
    player.Health.OnDie -= (health) => HandlePlayerDie(player);
}

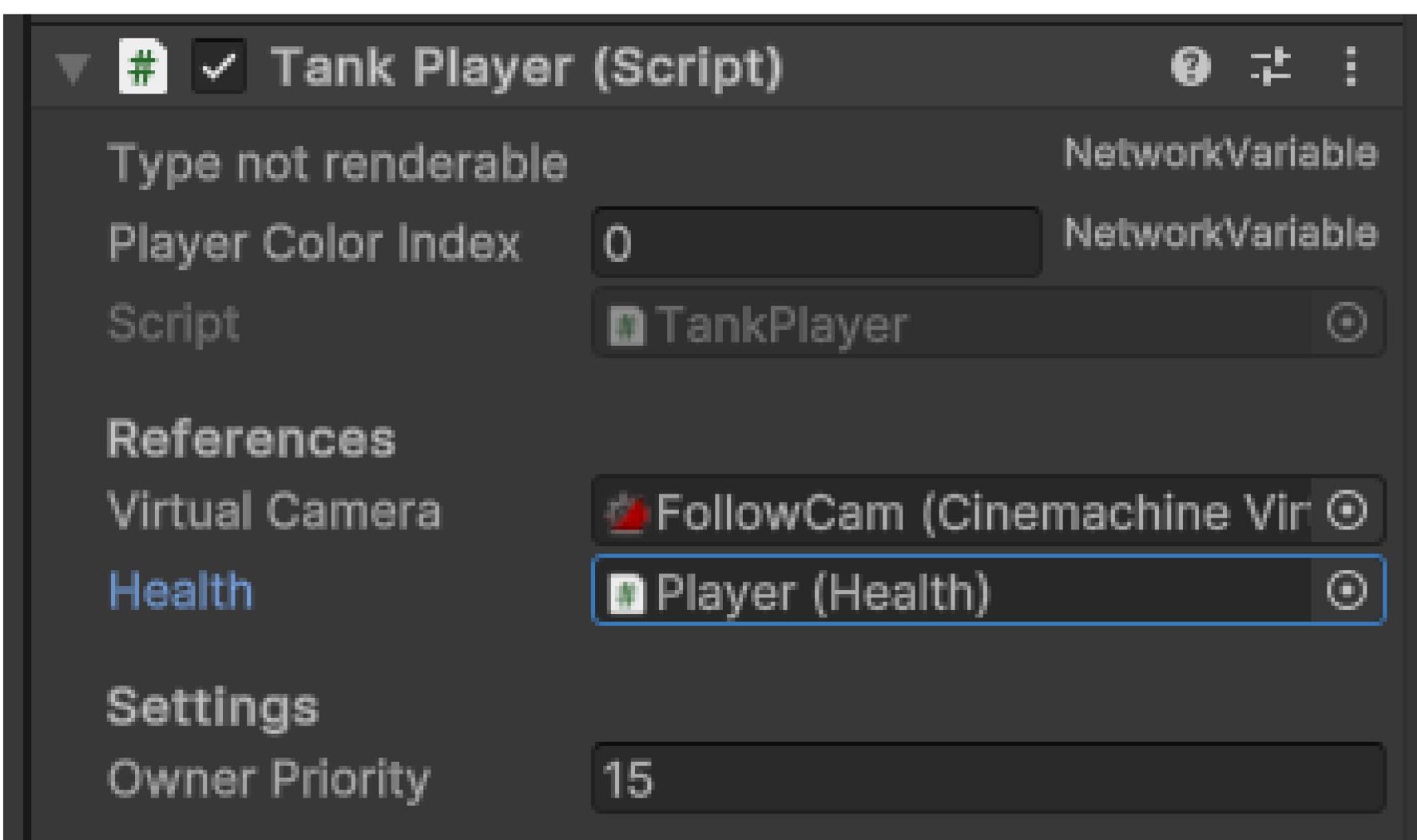
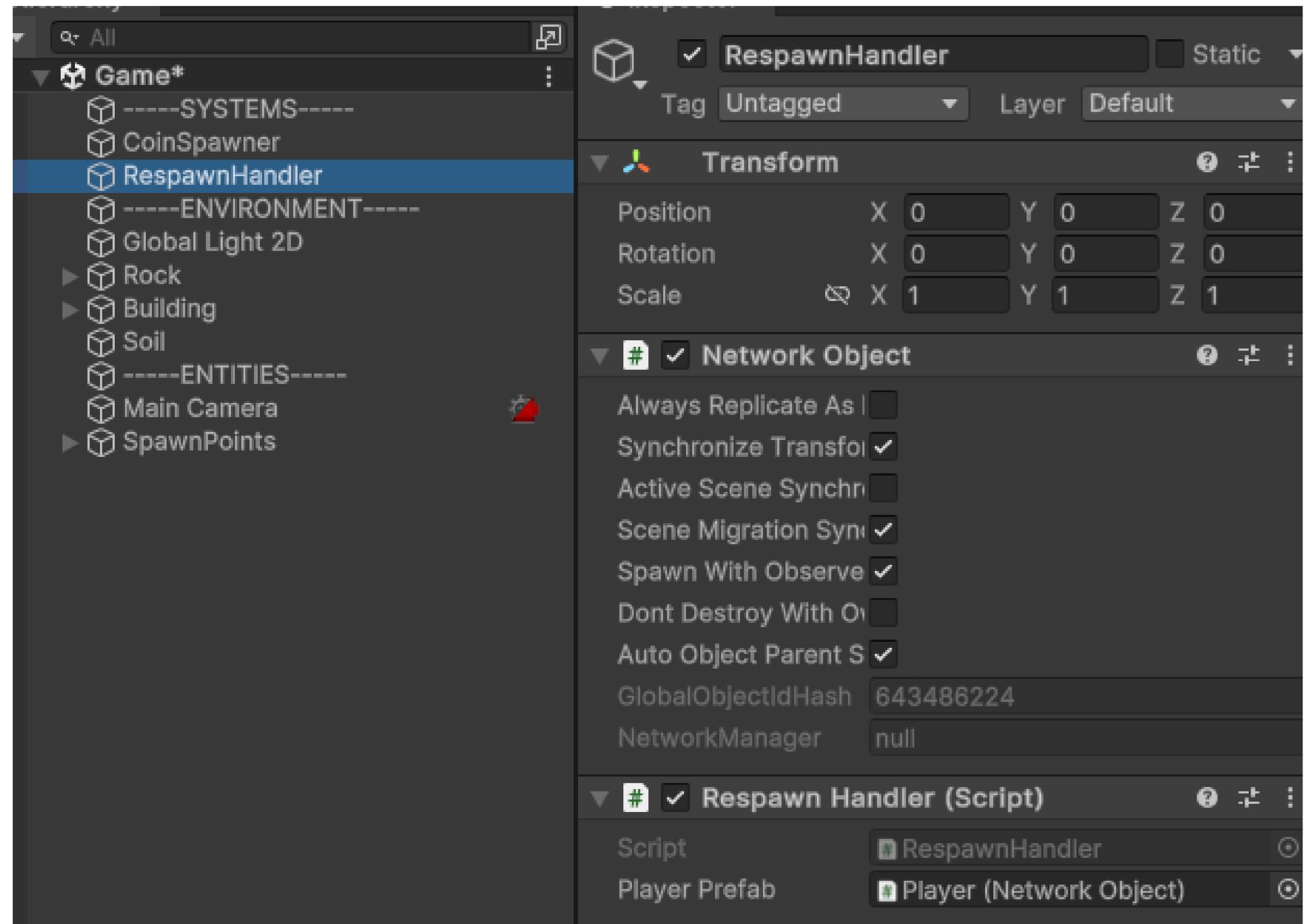
2 references
private void HandlePlayerDie(TankPlayer player)
{
    Destroy(player.gameObject);

    StartCoroutine(RespawnPlayer(player.ownerClientId));
}

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

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

    playerInstance.SpawnAsPlayerObject(ownerClientId);
}
```



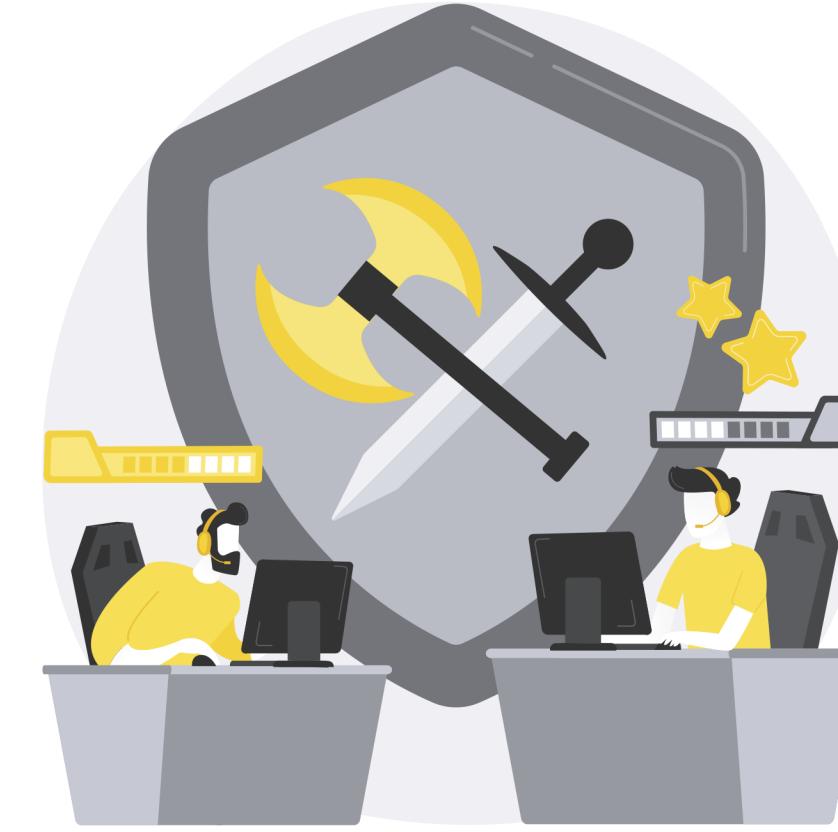


# Assignment

ให้ทำการด้วยคลิปผลลัพธ์ของการทำ Workshop ตามเนื้อหา Workshop ໃນແຕ່ລະ  
หัวข้อนี้พร้อมອธิบายประกอบ

- Player Camera
- Overhead Names
- Change Color
- Spawn Points
- Respawning

# Next Week



## Presentation Update 2

(Present your Progress Update Multiplayer Final Project)





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

**WOLVEDEN ACADEMY**



[facebook.com/GameDevMew](https://facebook.com/GameDevMew)