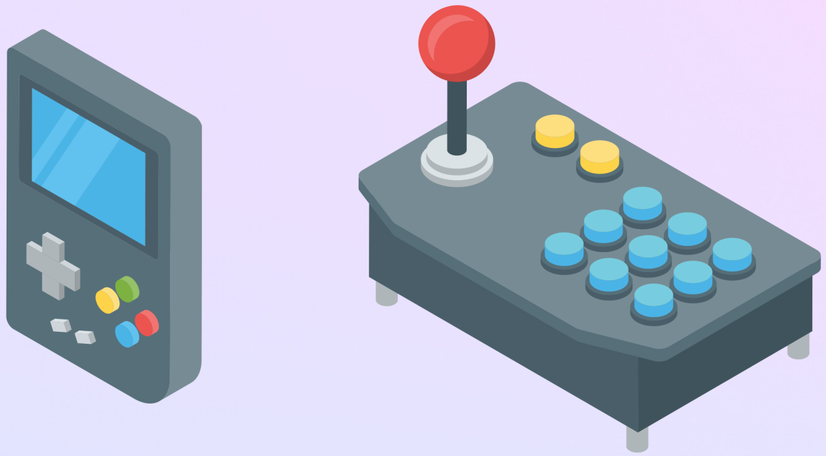


WOLVEDEN ACADEMY

NETWORKING AND MULTIPLAYER ONLINE GAMES

MULTIPLAYER NEW ERA

BY PONGSATHORN KIATTICHAROENPORN (MEW)

WEEK 4 : SAMPLE GAMEPLAY FOR MULTIPLAYER 1

Sample Gameplay for Multiplayer 1

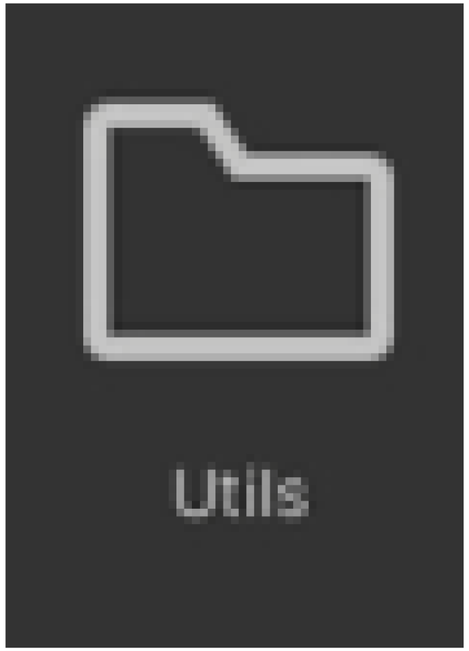
Topic in this week

Workshop

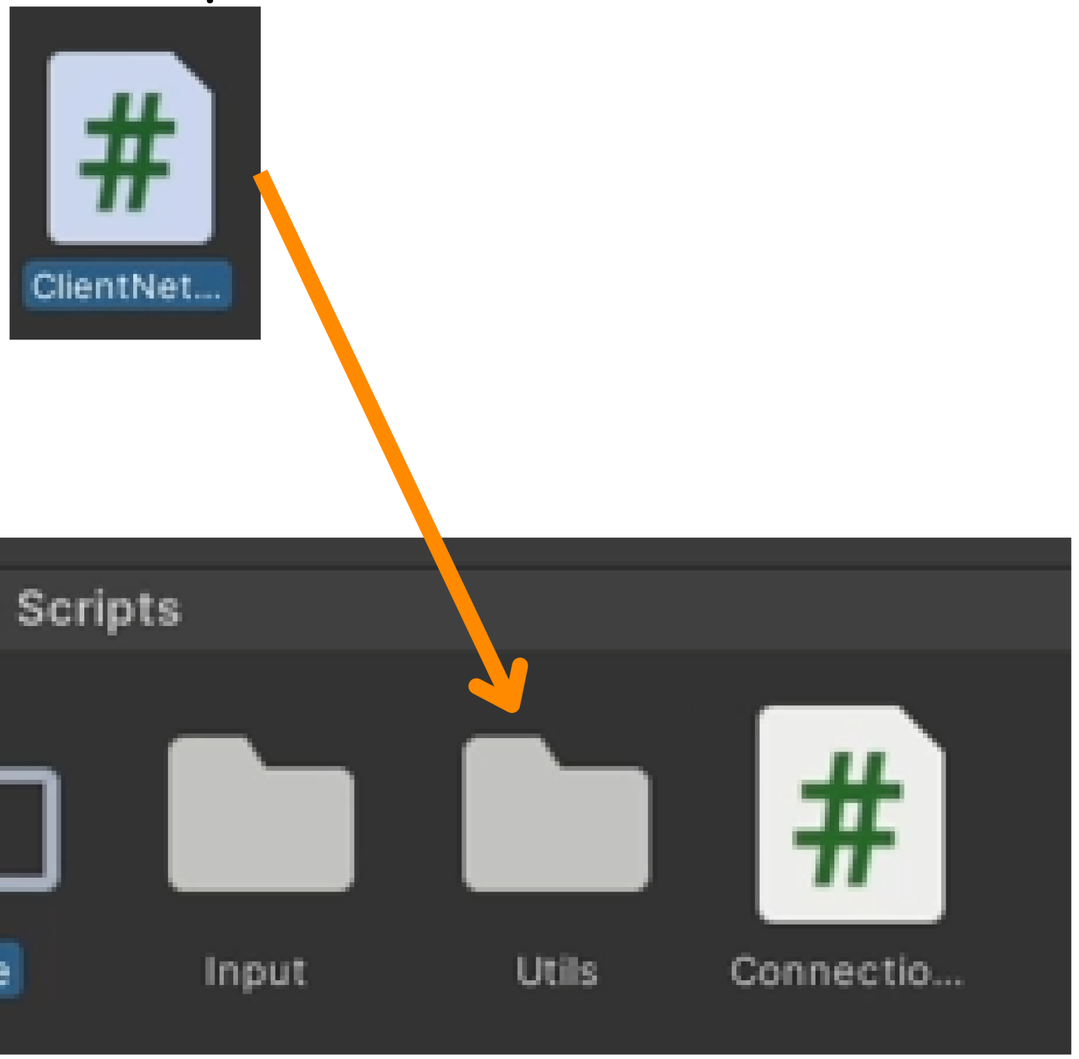
- Player Movement
- Player Aiming
- Networked Projectiles
- Firing Projectiles
- Firing Improvements

Player Movement

สร้างโฟลเดอร์

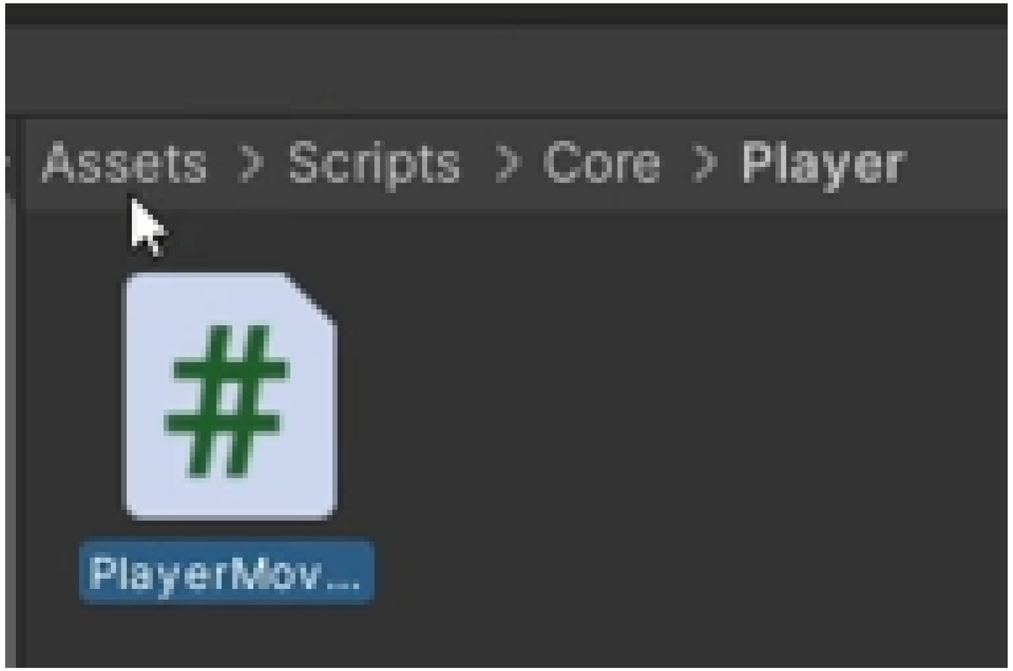


ย้าย Script ไปโฟลเดอร์ Utils



สร้างโฟลเดอร์ Player

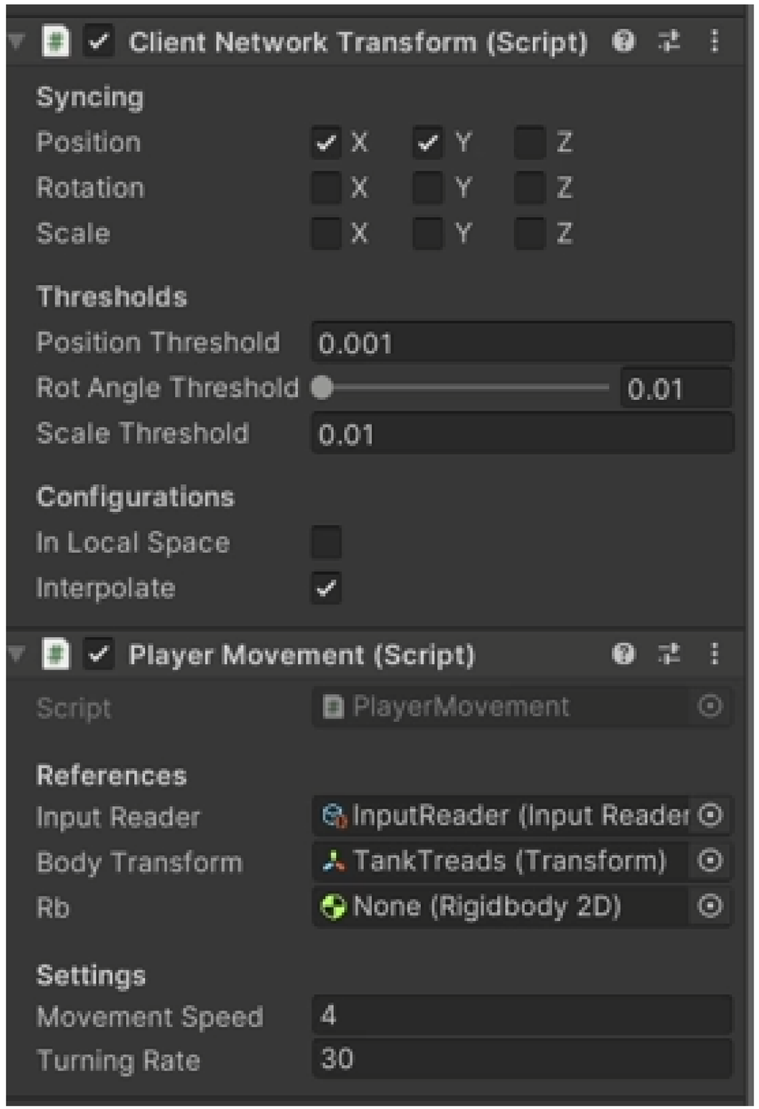
สร้าง Script ชื่อ PlayerMovement



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

Unity Script (1 asset reference) | 0 references
public class PlayerMovement : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private InputReader inputReader;
    [SerializeField] private Transform bodyTransform;
    [SerializeField] private Rigidbody2D rb;

    [Header("Settings")]
    [SerializeField] private float movementSpeed = 4f;
    [SerializeField] private float turningRate = 30f;
```

เขียนโค้ด PlayerMovement ส่วนแรก

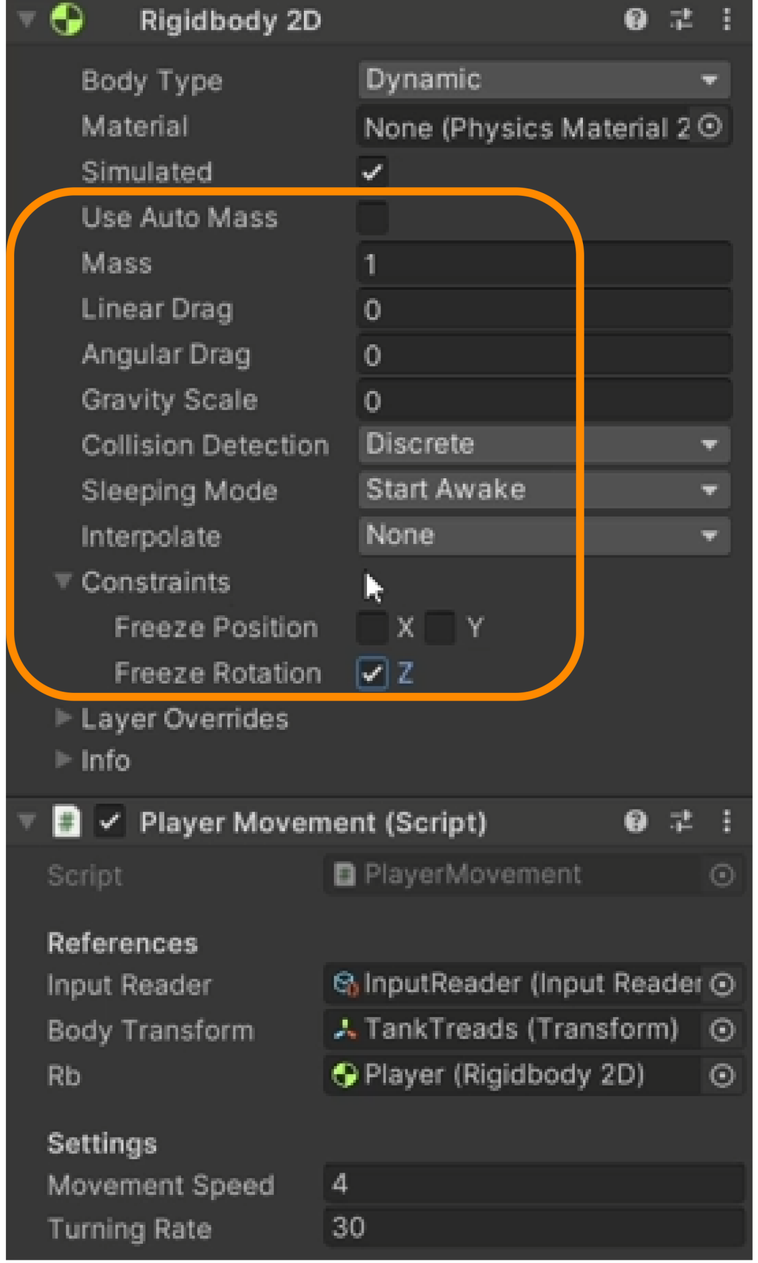
เอา Script PlayerMovement.cs มา add
ใน prefab Player



*หมายเหตุ ตัวที่จะเอามาใส่ใน BodyTransform จำเป็นต้องมี Client Network Transform อยู่ด้วย เพื่อ sync ผ่าน network

เพิ่ม Rigidbody2D เข้ามาใน Player

โดยตั้งค่าตามในภาพ



แล้วลากตัว Player เองเข้าไปใส่ใน Rb

เขียนโค้ด PlayerMovement ส่วนที่

```
public override void OnNetworkSpawn()
{
    if (!IsOwner) { return; }



}

0 references
public override void OnNetworkDespawn()
{
    if (!IsOwner) { return; }



}
```

*__หมายเหตุ__ ต้อง Inherit เป็น NetworkBehaviour และ using Unity.Netcode ด้วยถึงจะ เรียก override 2 function นี้ได้

Read the Movement Input

- Create a local **Vector2 previousMovementInput**
- Create a **HandleMove(Vector2)** method
- Subscribe to the MoveEvent in **OnNetworkSpawn()**
- Unsubscribe in **OnNetworkDespawn()**
- Write to the local variable

- Create a local **Vector2 previousMovementInput**
  - Create a **HandleMove(Vector2)** method
  - Subscribe to the MoveEvent in **OnNetworkSpawn()**
  - Unsubscribe in **OnNetworkDespawn()**
  - Write to the local variable

```
Unity Script (1 asset reference) | 0 references
public class PlayerMovement : NetworkBehaviour
{
    [Header("References")]
    [SerializeField] private InputReader inputReader;
    [SerializeField] private Transform bodyTransform;
    [SerializeField] private Rigidbody2D rb;

    [Header("Settings")]
    [SerializeField] private float movementSpeed = 4f;
    [SerializeField] private float turningRate = 30f;

    private Vector2 previousMovementInput;
```

- Create a **HandleMove(Vector2)** method
  - Subscribe to the MoveEvent in **OnNetworkSpawn()**
  - Unsubscribe in **OnNetworkDespawn()**
  - Write to the local variable

```
private void HandleMove(Vector2 movementInput)
{
    previousMovementInput = movementInput;
}
```

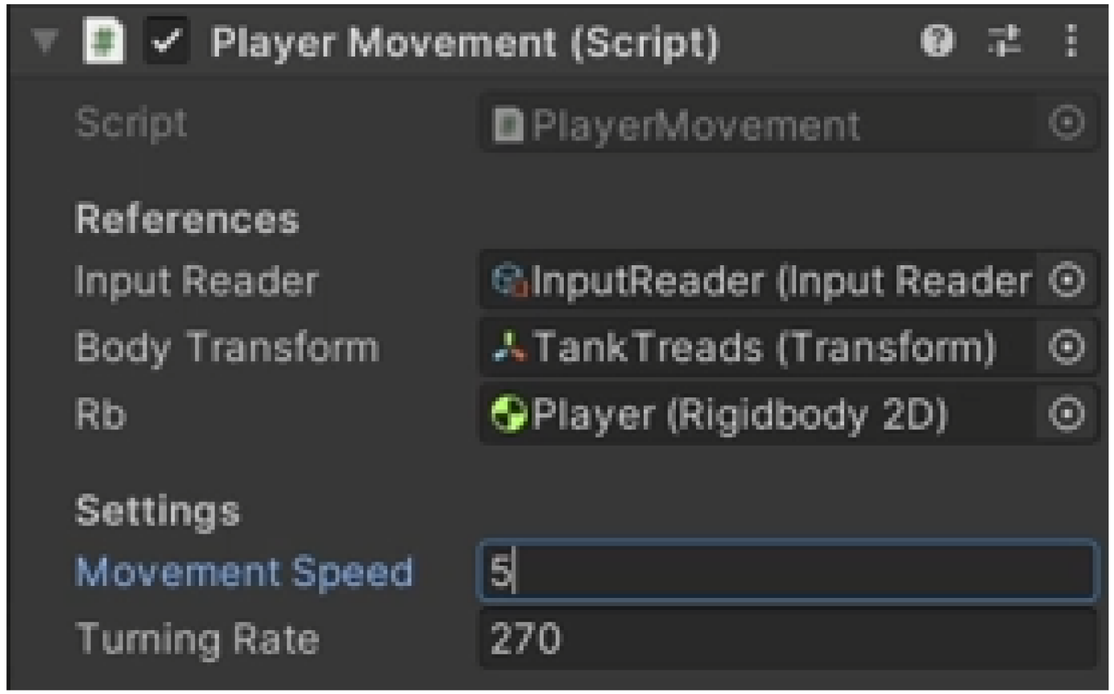
- Subscribe to the MoveEvent in **OnNetworkSpawn()**
- Unsubscribe in **OnNetworkDespawn()**
- Write to the local variable

```
0 references
public override void OnNetworkSpawn()
{
    if (!IsOwner) { return; }
    inputReader.MoveEvent += HandleMove;
}
0 references
public override void OnNetworkDespawn()
{
    if(!IsOwner) { return; }
    inputReader.MoveEvent -= HandleMove;
}
```

- Write to the local variable (1/2)

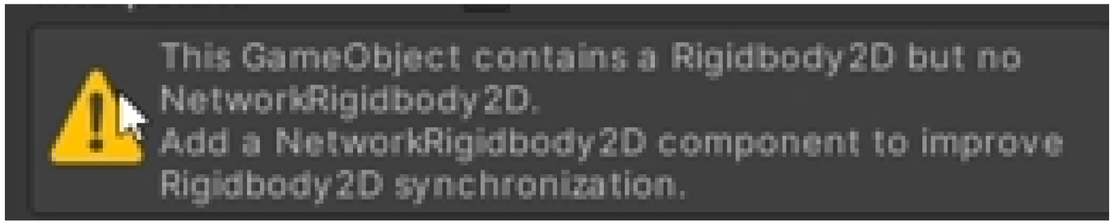
```
void Update()
{
    if(!IsOwner) { return ; }

    float zRotation = previousMovementInput.x * -turningRate * Time.deltaTime;
    bodyTransform.Rotate(0f,0f,zRotation);
}
```

Time to Test Turning Tank

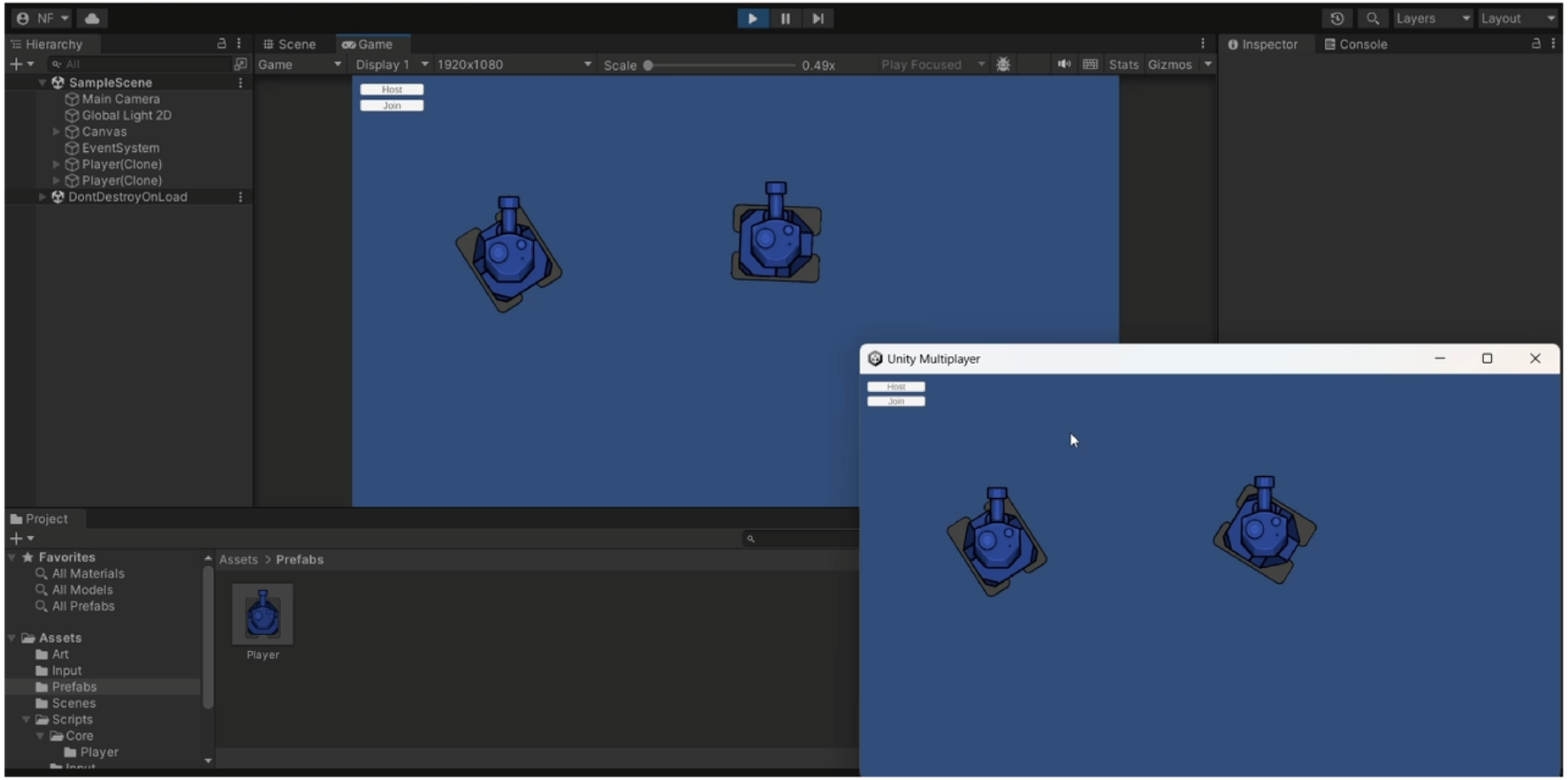


- Write to the local variable (2/2)

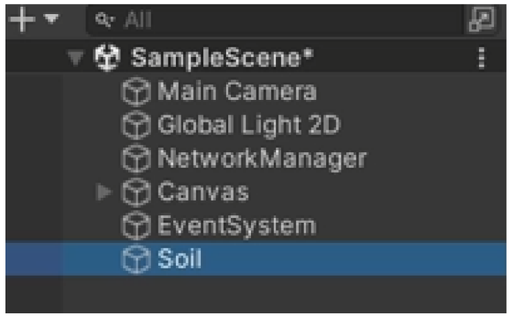
```
private void FixedUpdate()
{
    if (!IsOwner) { return ; }

    rb.velocity = (Vector2)bodyTransform.up * previousMovementInput.y * movementSpeed;
}
```

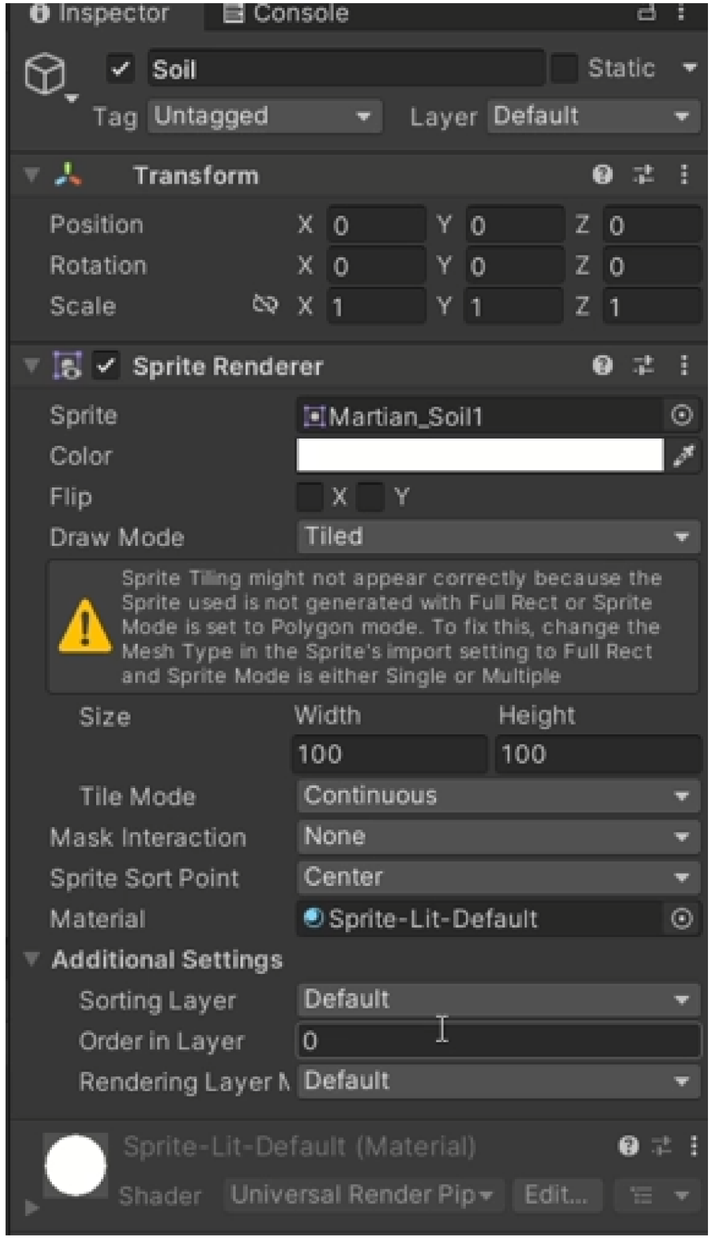

This GameObject contains a Rigidbody2D but no NetworkRigidbody2D.
Add a NetworkRigidbody2D component to improve Rigidbody2D synchronization.

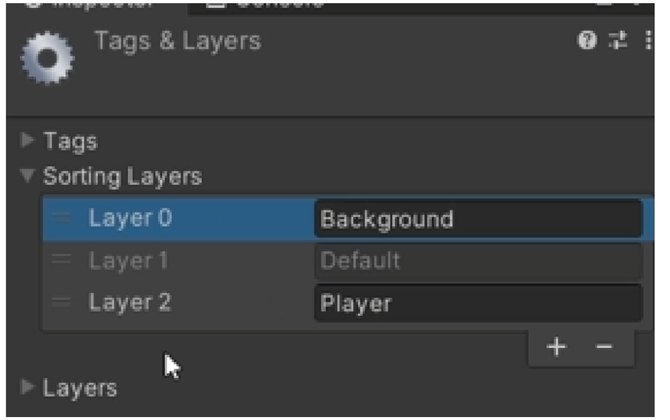
Time to Test Moving & Turning Tank

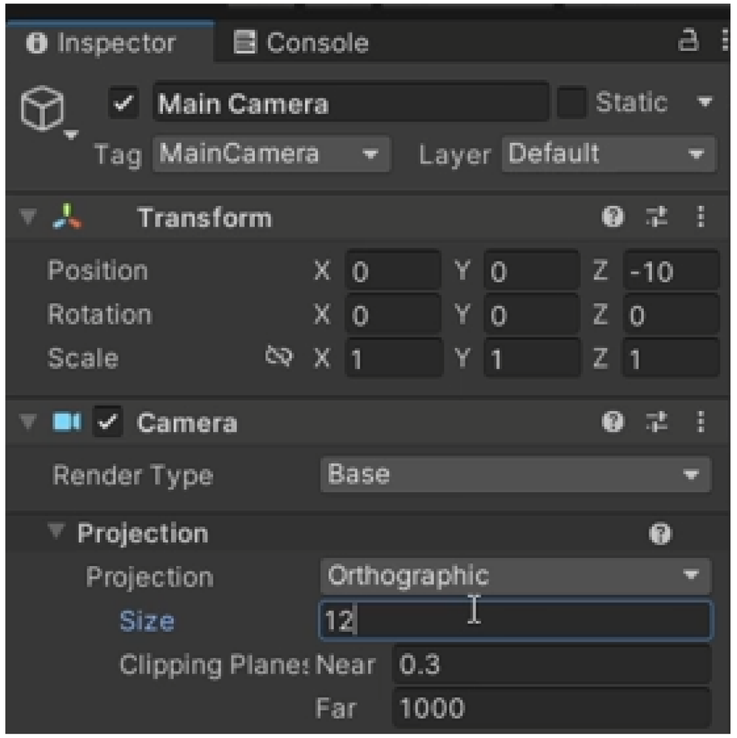


ขั้นตอนการเพิ่ม Background ฉาก

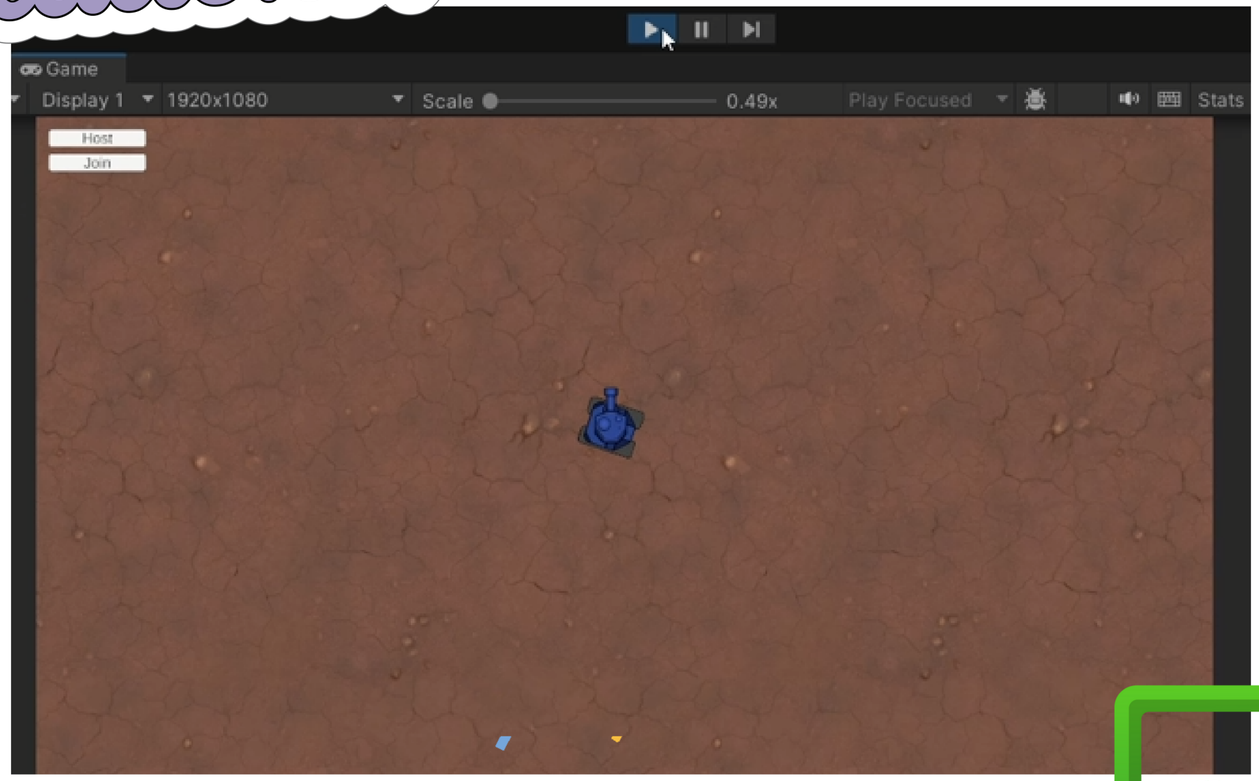






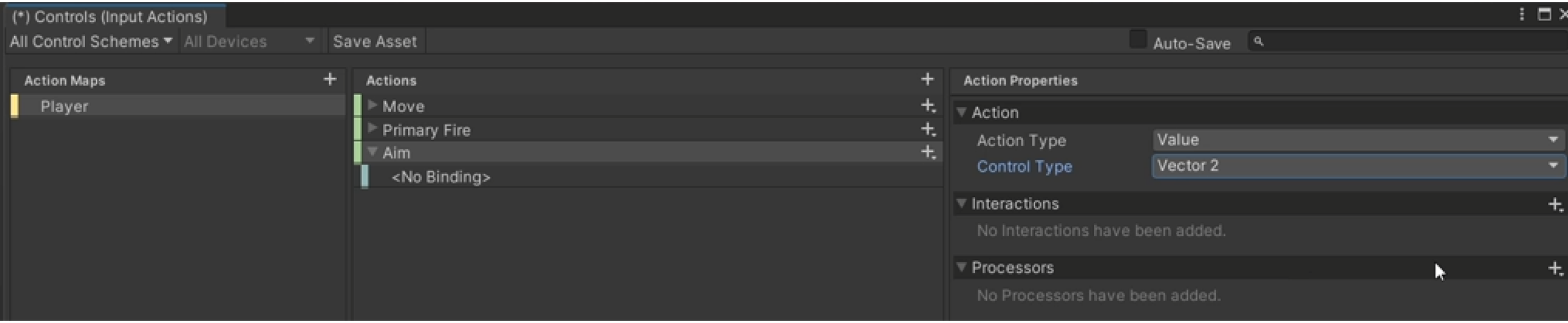


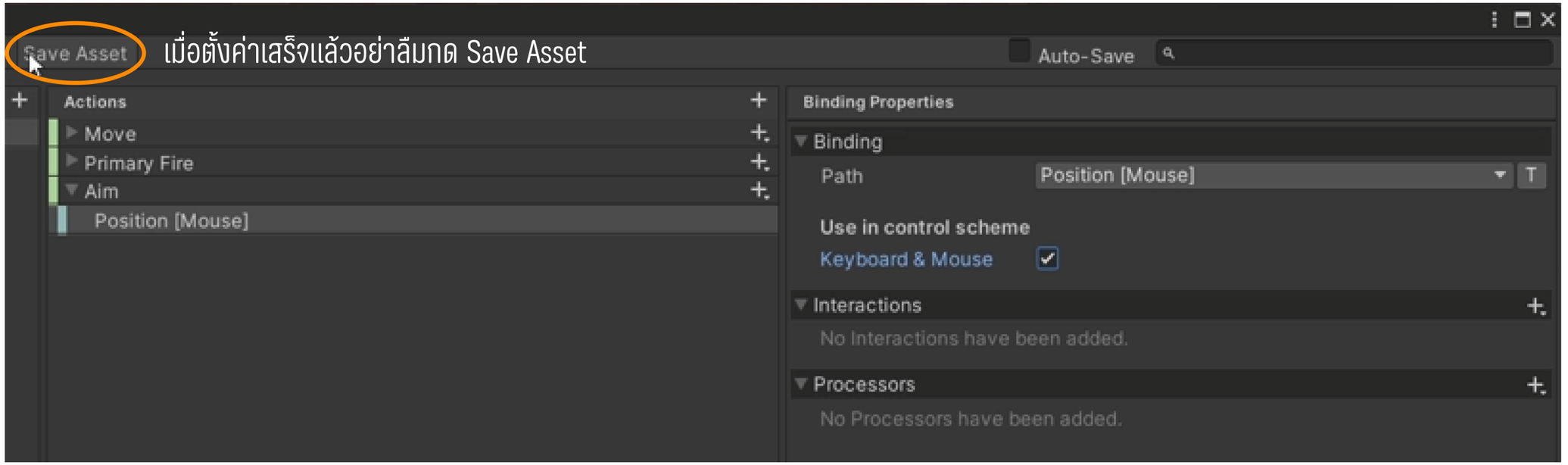
Player Movement

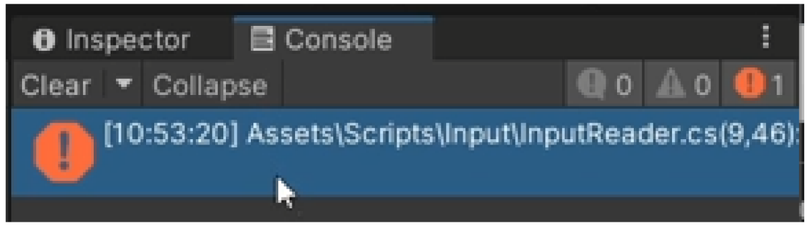


Player Aiming

เพิ่ม Input Actions “Aim” โดยตั้งค่าตามนี้








```
Assets\Scripts\Input\InputReader.cs(9,46): error CS0535: 'InputReader' does not implement interface member 'Controls.IPlayerActions.OnAim(InputAction.CallbackContext)'
```

```
interface Controls.IPlayerActions

'InputReader' does not implement interface member
'Controls.IPlayerActions.OnAim(InputAction.CallbackContext)'
[Assembly-CSharp] csharp(CS0535)

View Problem (Alt+F8)    Quick Fix... (Ctrl+.)
```

```
using static Controls;

[CreateAssetMenu(fileName = "New Input Reader", menuName = "Input/Input Reader")]
1 reference
public class InputReader : ScriptableObject, IPlayerActions
{
```

```
IPlayerActions
Quick Fix...
Implement interface
```

```
public void OnAim(InputAction.CallbackContext context)
{
    throw new NotImplementedException();
}
```

```
public void OnAim(InputAction.CallbackContext context)
{
    context.ReadValue<Vector2>();
}
```

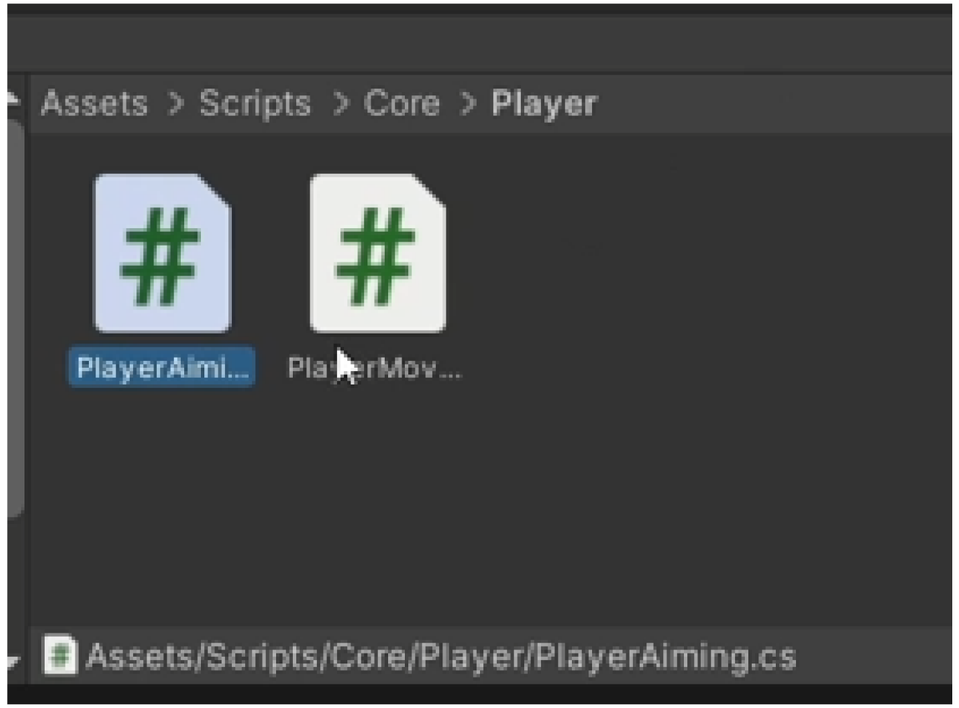
InputReader.cs | PlayerMovement.cs | ProjectileLauncher.cs

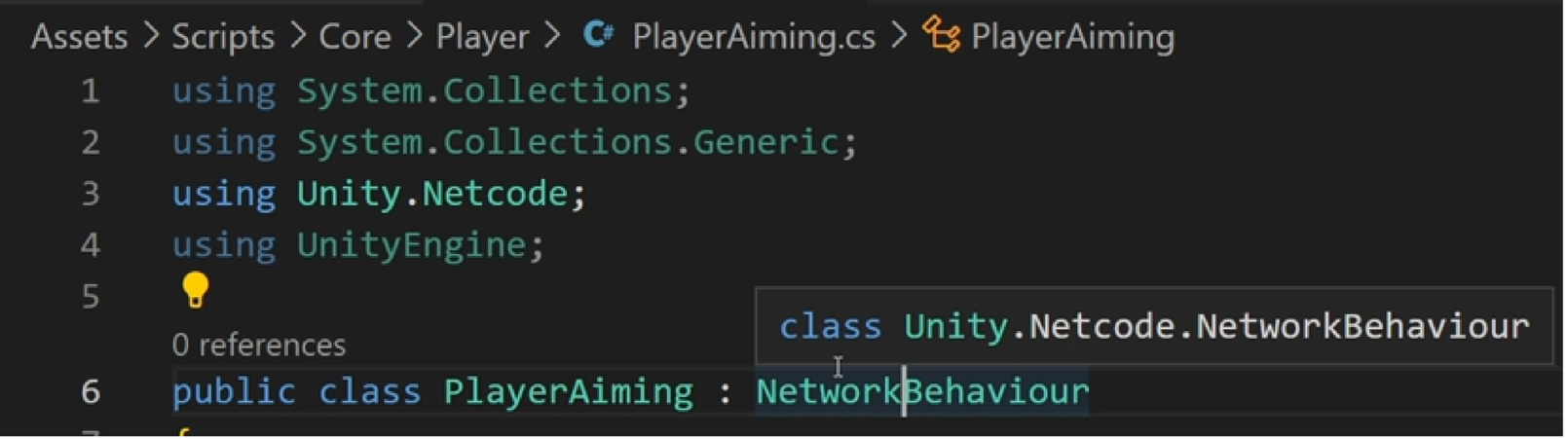
Assembly-CSharp | InputReader

```
1   using System;
2   using System.Collections;
3   using System.Collections.Generic;
4   using UnityEngine;
5   using UnityEngine.InputSystem;
6   using static Controls;
7
8   [CreateAssetMenu(fileName = "New Input Reader", menuName = "Input/Input Reader")]
        Unity Script | 3 references
9   public class InputReader : ScriptableObject, IPlayerActions
10  {
11      public event Action<Vector2> MoveEvent;
12      public event Action<bool> PrimaryFireEvent;
13
            2 references
14      public Vector2 AimPosition { get; private set; }
15
16      private Controls controls;
```

```
public void OnAim(InputAction.CallbackContext context)
{
    AimPosition = context.ReadValue<Vector2>();
}
```

สร้าง Script ใหม่ PlayerAiming



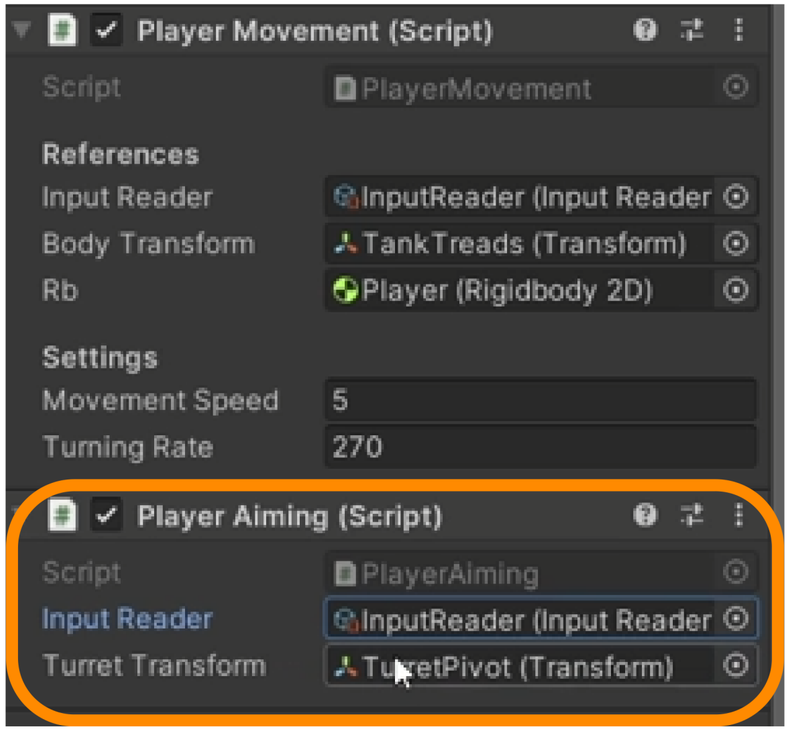



```
0 references
public class PlayerAiming : NetworkBehaviour
{
    0 references
    [SerializeField] private InputReader inputReader;
    0 references
    [SerializeField] private Transform turretTransform;

    0 references
    private void LateUpdate()
    {

    }
}
```

Player Aiming

- Make sure we are the owner
- Get the aim position
- **Camera.main.ScreenToWorldPoint(Vector2)**
- Set the turret's up vector to point at the cursor (the difference between the cursor and turret position)

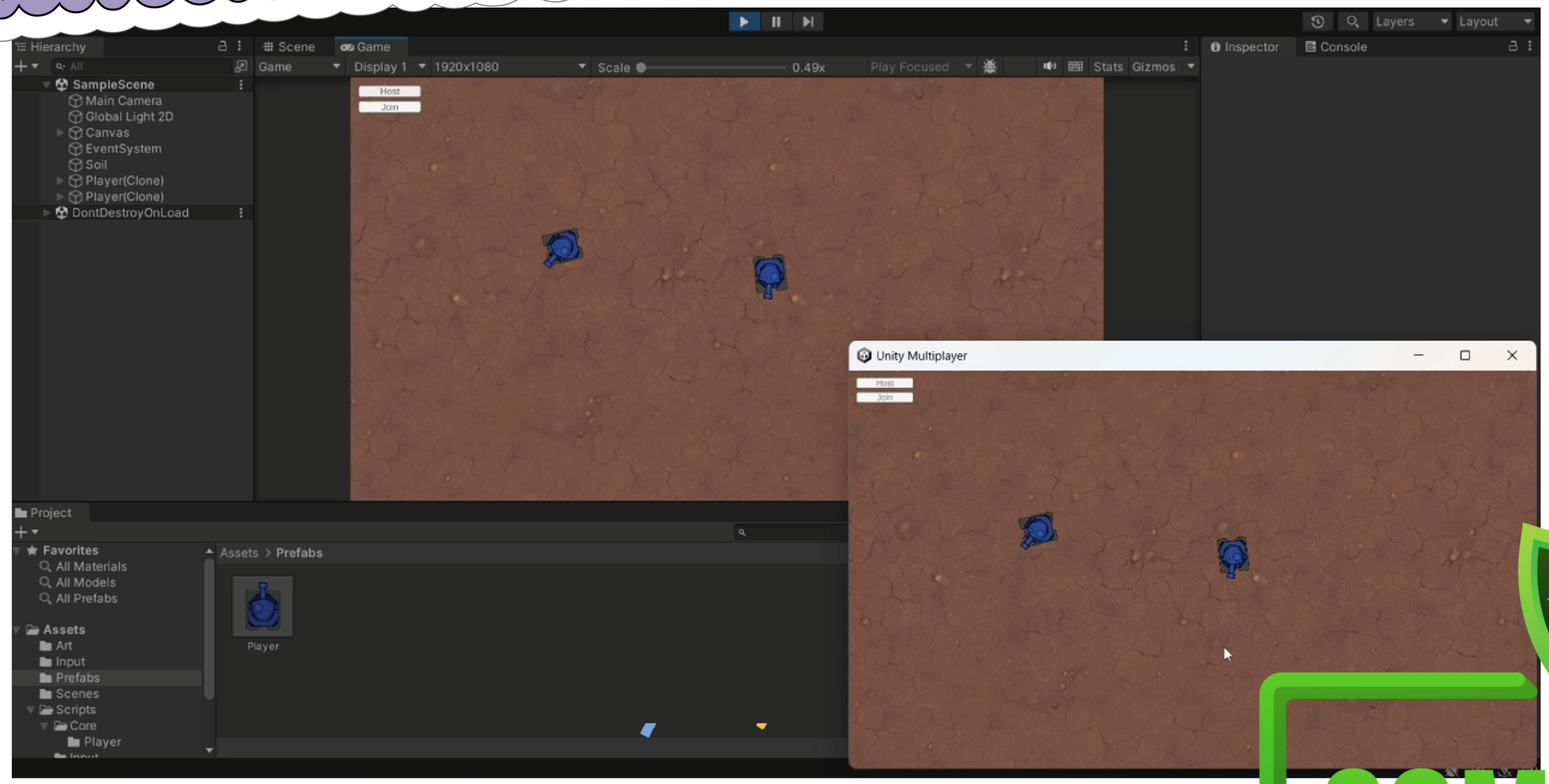
```
private void LateUpdate()
{
    if (!IsOwner) { return; }

    Vector2 aimScreenPosition = inputReader.AimPosition;
    Vector2 aimWorldPosition = Camera.main.ScreenToWorldPoint(aimScreenPosition);

    turretTransform.up = new Vector2(
        aimWorldPosition.x - turretTransform.position.x,
        aimWorldPosition.y - turretTransform.position.y);
}
```



*หมายเหตุ ตัวที่จะเอามาใส่ใน TurretTransform จำเป็นต้องมี Client Network Transform อยู่ด้วย เพื่อ sync ผ่าน network

Time to Test “Player Aiming”

Player Aiming



Networked Projectiles

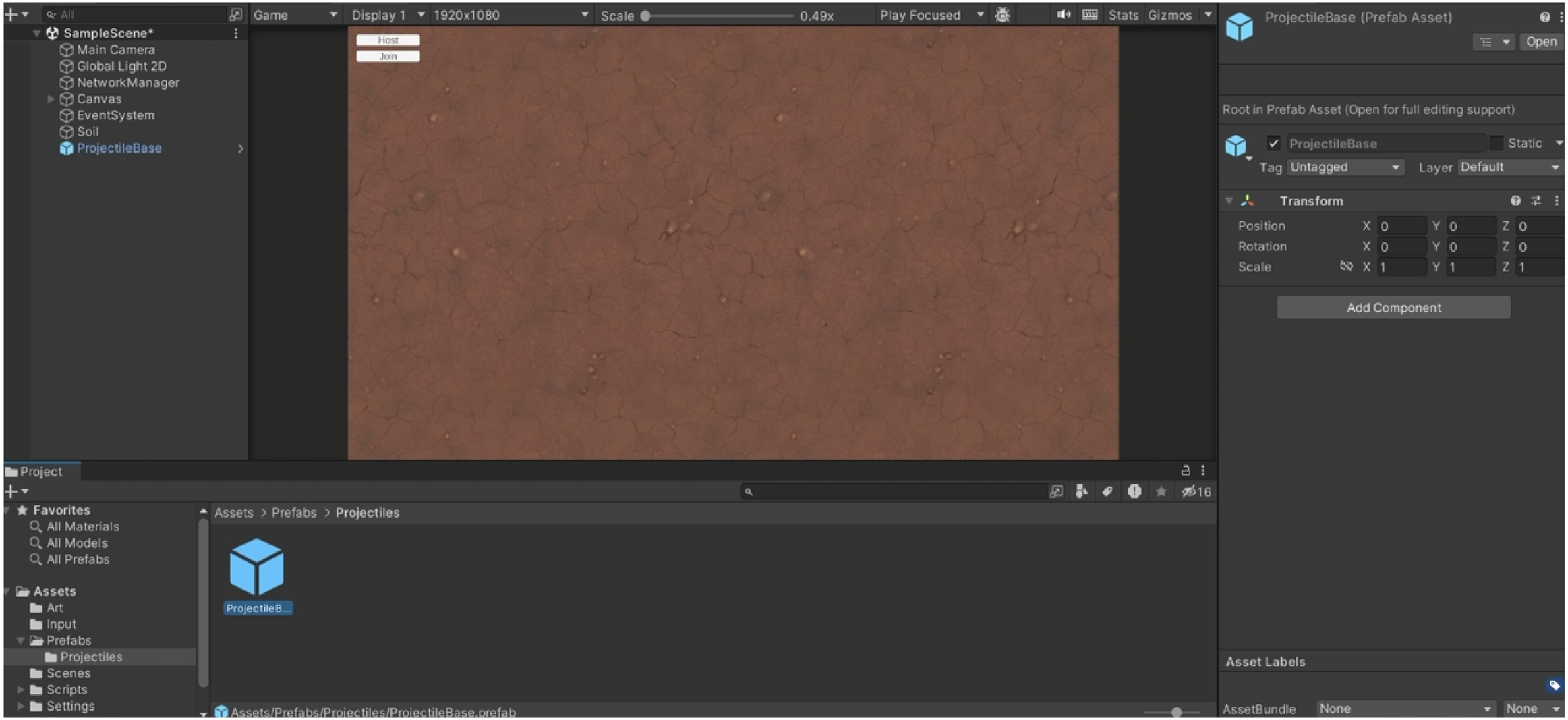
Order Of Events

1. Dummy projectile spawned for the firing client
2. RPC sent to server to try and fire
3. Server validates and then spawns the “real” projectile
4. Server sends RPC to clients to spawn dummy projectiles

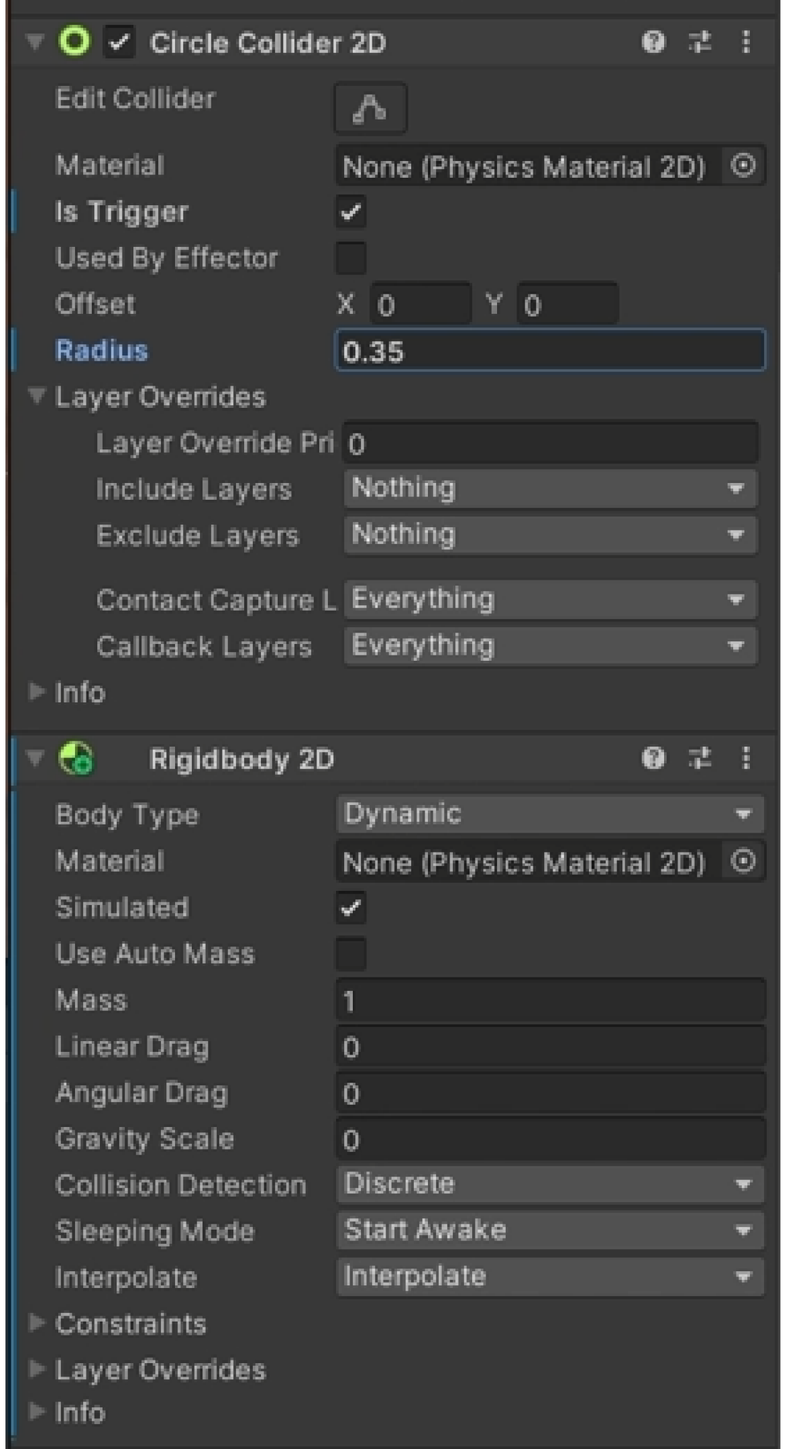
RPC: Remote Procedure Call

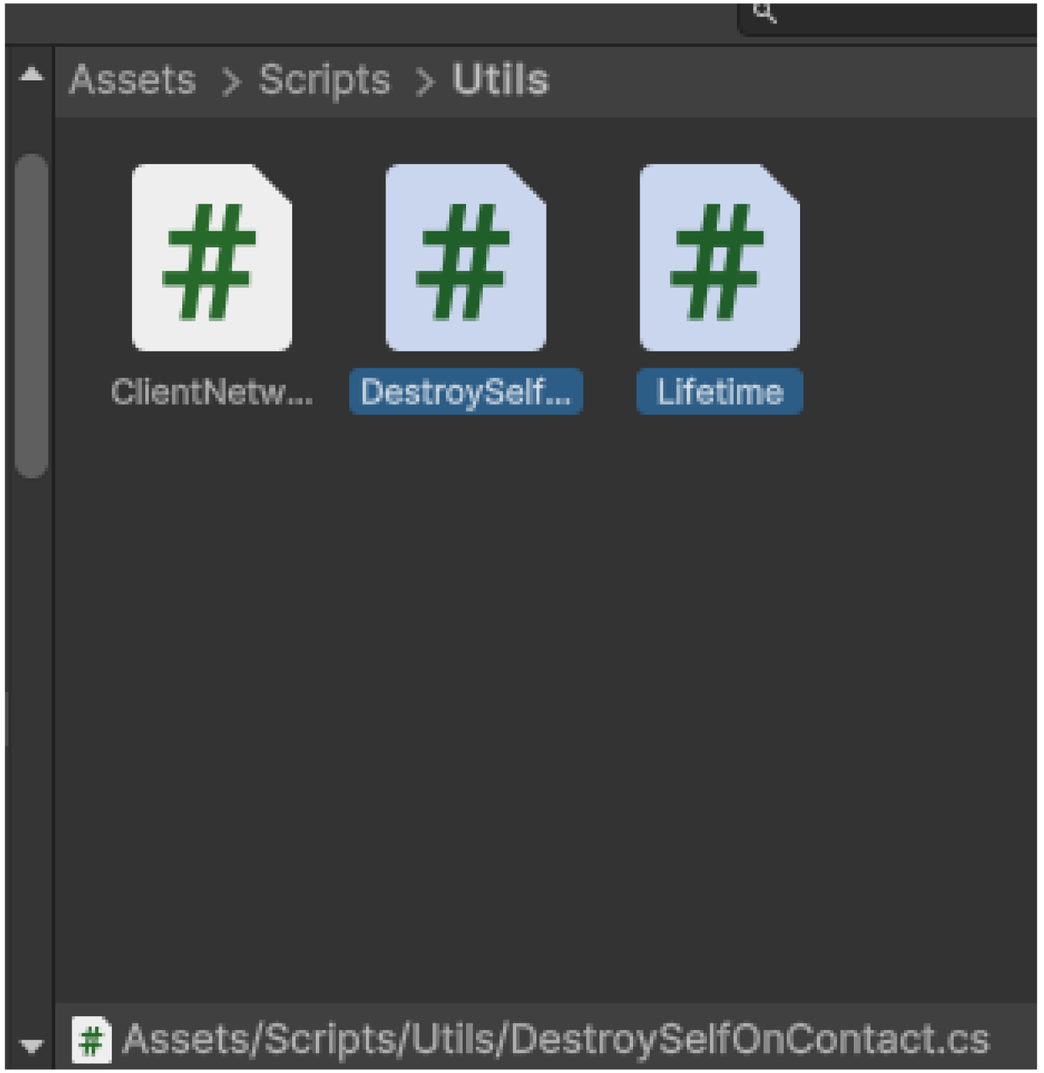
The concept of an RPC is common not only in video games but in the software industry in general. They're ways to call methods on objects that aren't in the same executable.

สร้าง Folder ชื่อ “Projectiles” และ Prefab ชื่อ “ProjectileBase”



เพิ่ม Component “Circle Collider 2D” และ “Rigidbody 2D”
โดยตั้งค่าตามนี้




Assets > Scripts > Utils
ClientNetw...
DestroySelf...
Lifetime
Assets/Scripts/Utils/DestroySelfOnContact.cs

Lifetime & Destroy On Contact

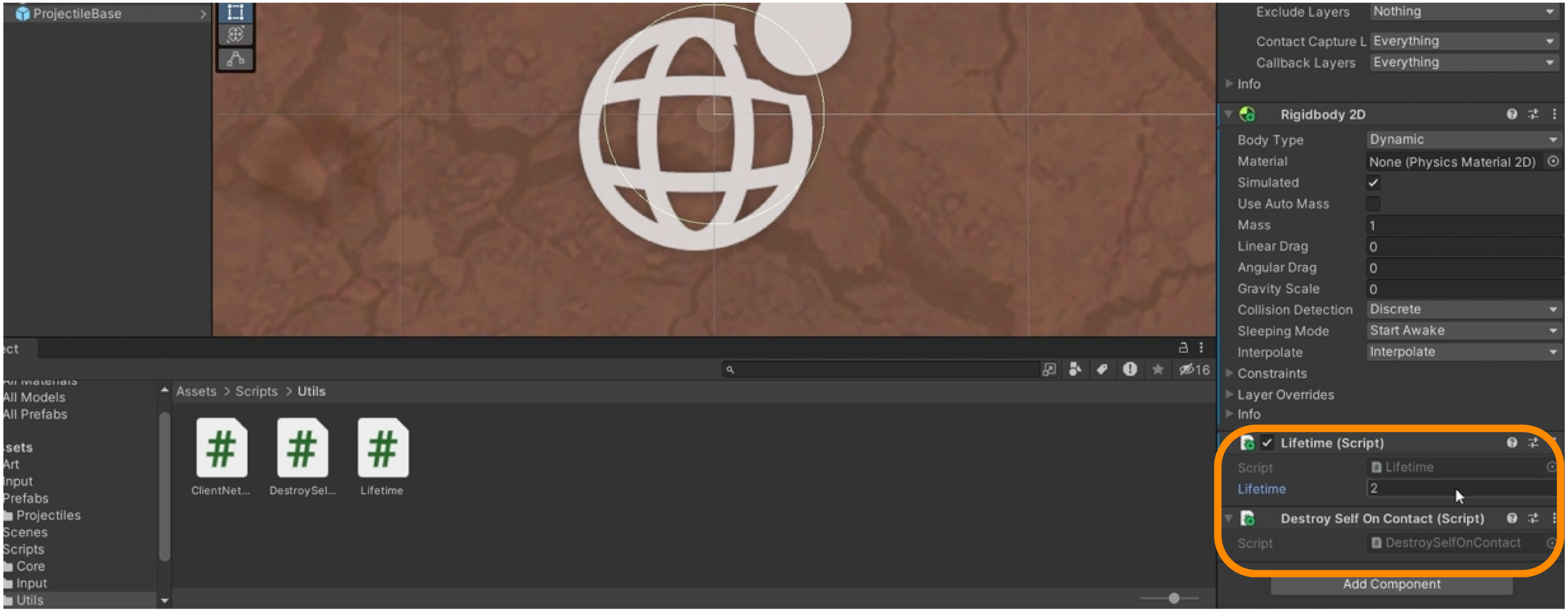
- 1st Script: Destroy the gameobject after x seconds
- 2nd Script: Destroy the gameobject when it hits another object using

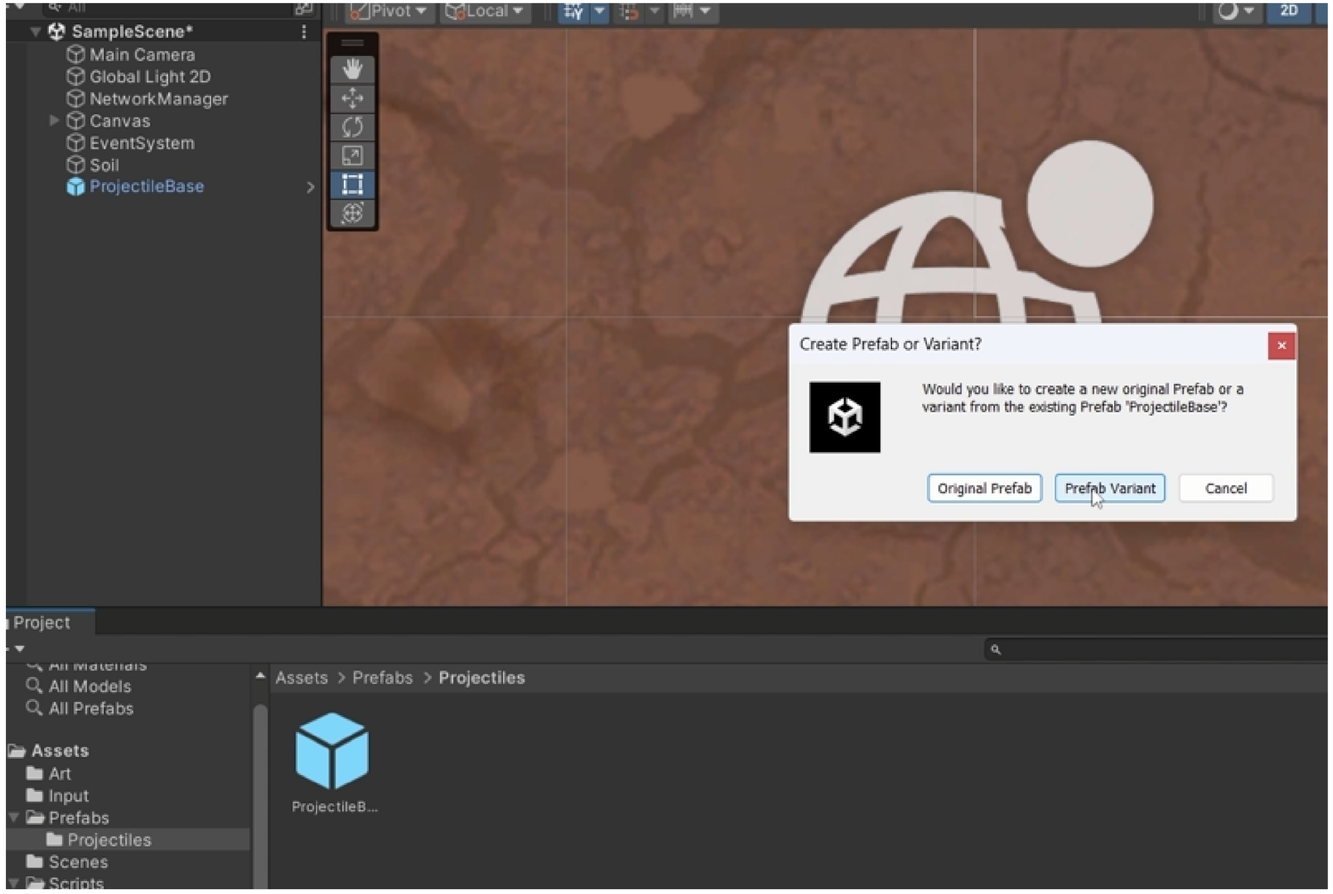
  OnTriggerEnter2D(Collider2D)

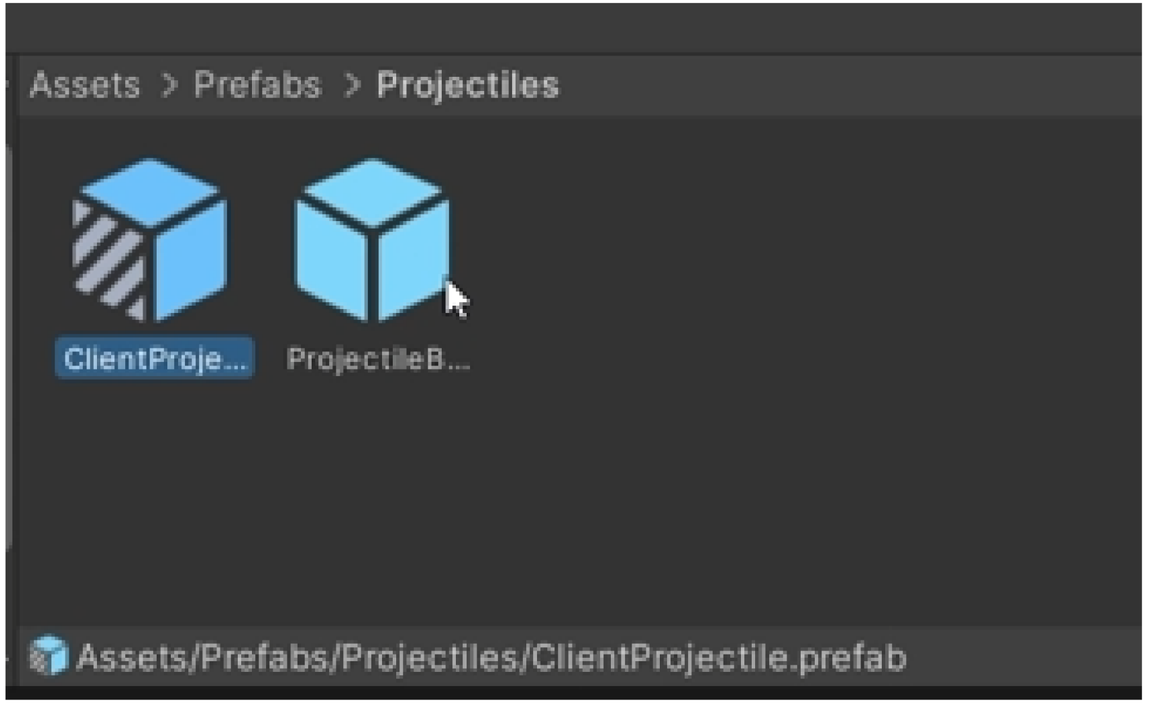
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class DestroySelfOnContact : MonoBehaviour
{
    Unity Message | 0 references
    private void OnTriggerEnter2D(Collider2D col)
    {
        Destroy(gameObject);
    }
}
```

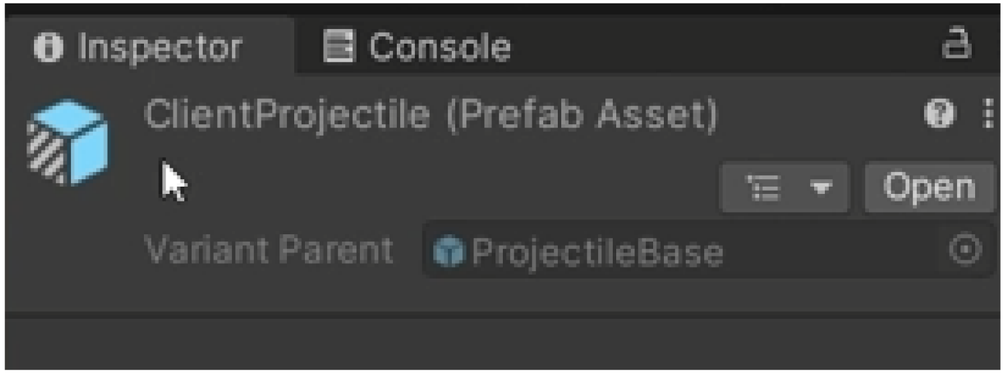
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class Lifetime : MonoBehaviour
{
    [SerializeField] private float lifetime = 1f;
    Unity Message | 0 references
    void Start()
    {
        Destroy(gameObject,lifetime);
    }

}
```

เพิ่ม Component “Lifetime” และ “DestorySelfOnContact” โดยตั้งค่าตามนี้



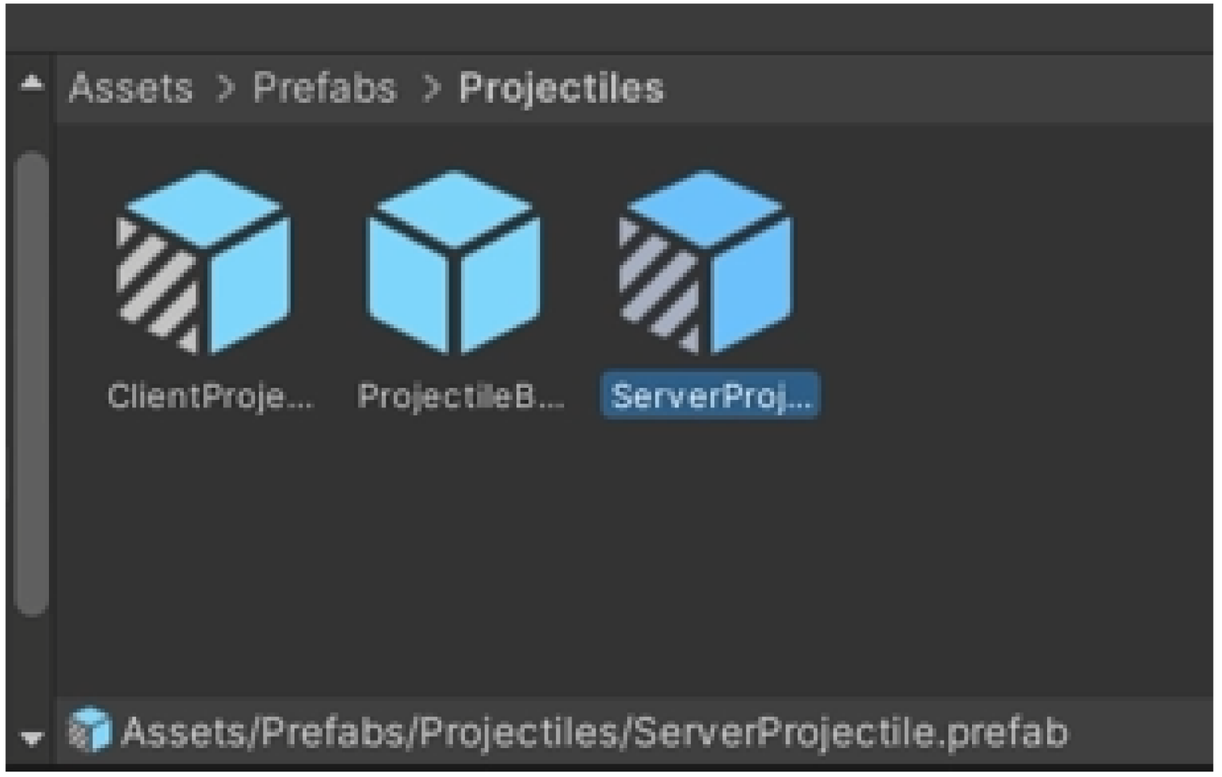

SampleScene*
Main Camera
Global Light 2D
NetworkManager
Canvas
EventSystem
Soil
ProjectileBase
Pivot
Local
2D
Create Prefab or Variant?
Would you like to create a new original Prefab or a variant from the existing Prefab 'ProjectileBase'?
Original Prefab
Prefab Variant
Cancel
Project
All Models
All Prefabs
Assets
Art
Input
Prefabs
Projectiles
Scenes
Scripts
Assets > Prefabs > Projectiles
ProjectileB...

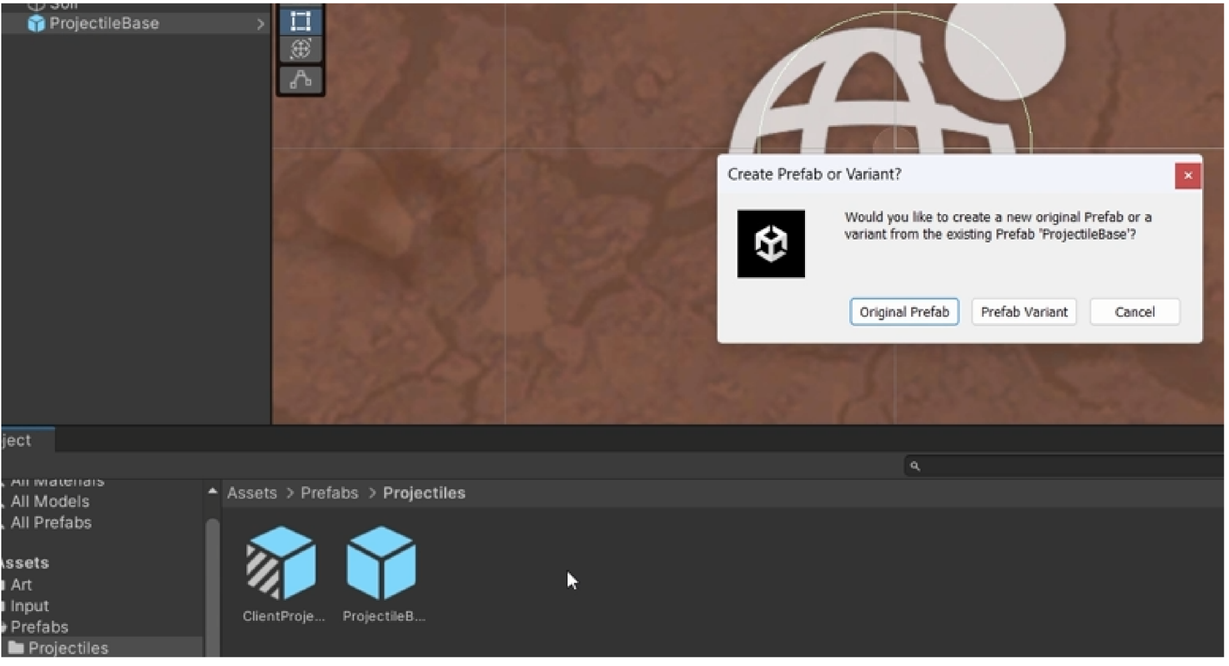

Assets > Prefabs > Projectiles
ClientProje...
ProjectileB...
Assets/Prefabs/Projectiles/ClientProjectile.prefab

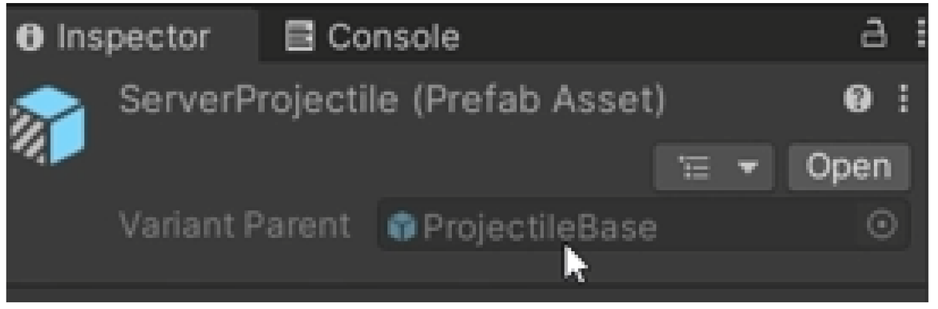

Inspector
Console
ClientProjectile (Prefab Asset)
Open
Variant Parent
ProjectileBase

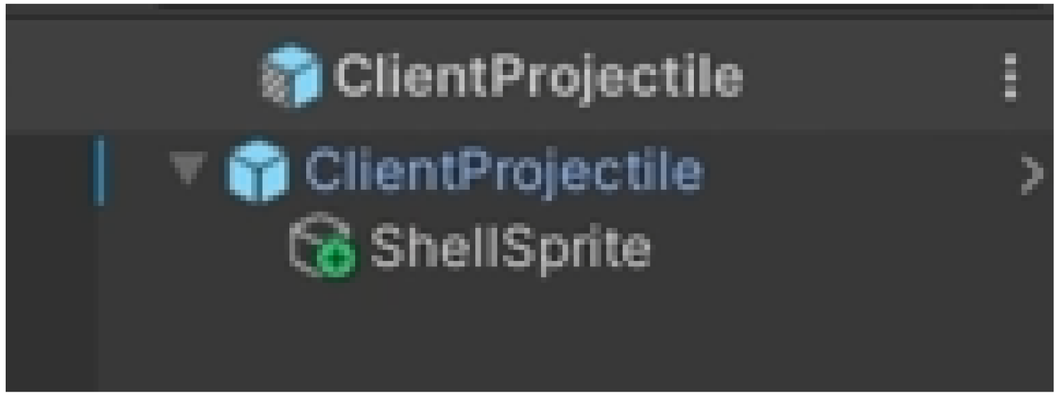

Soil
ProjectileBase

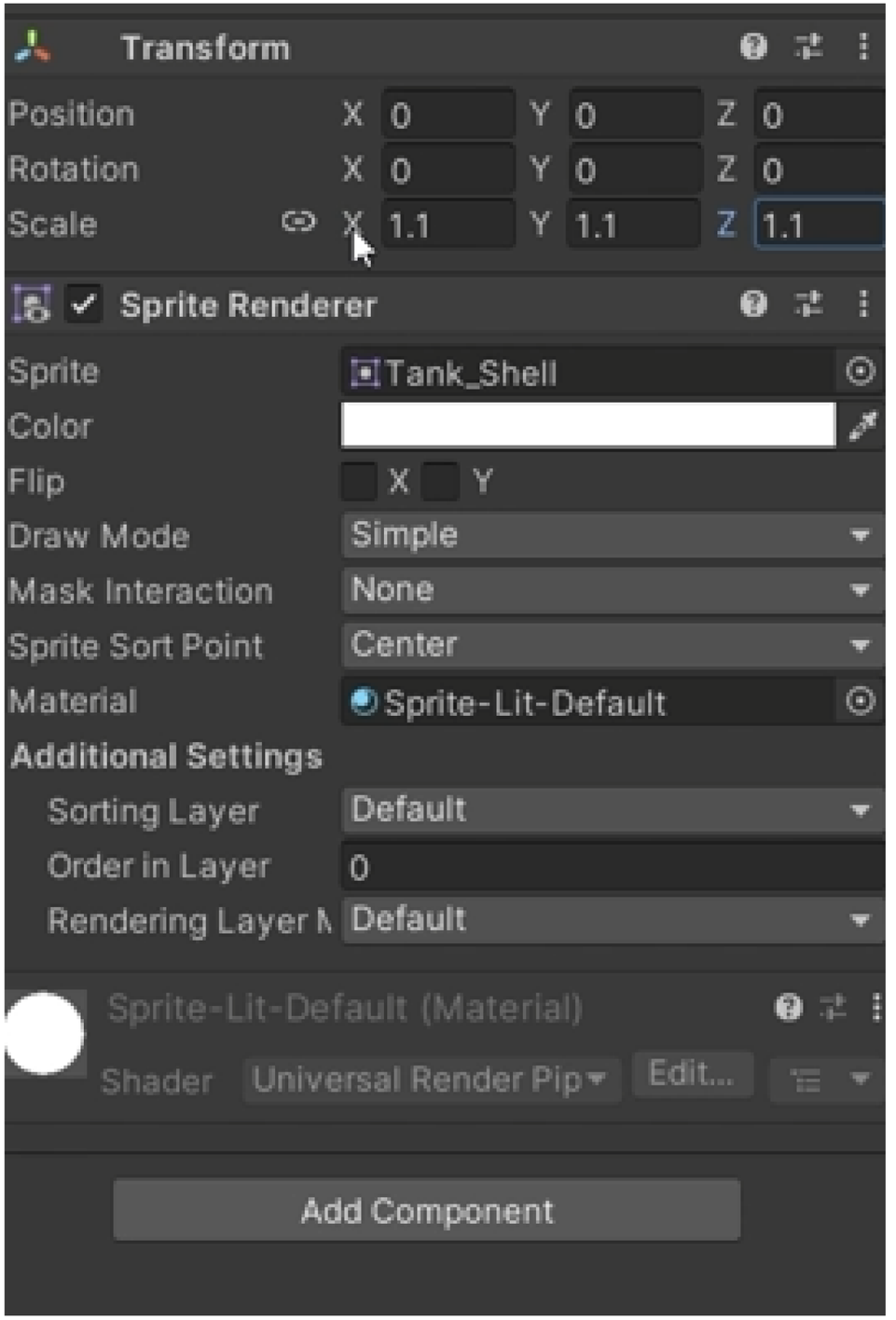

ProjectileBase


Assets > Prefabs > Projectiles
ClientProje...
ProjectileB...
ServerProj...
Assets/Prefabs/Projectiles/ServerProjectile.prefab

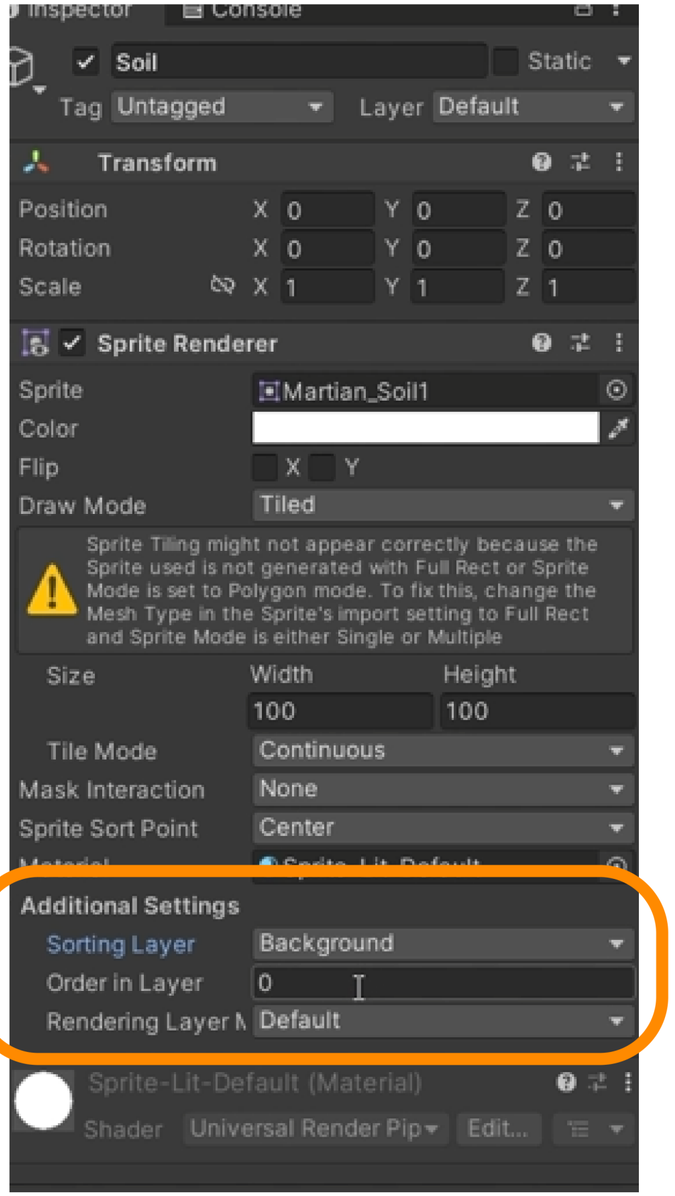

ProjectileBase
Create Prefab or Variant?
Would you like to create a new original Prefab or a variant from the existing Prefab 'ProjectileBase'?
Original Prefab
Prefab Variant
Cancel
All Models
All Prefabs
Assets
Art
Input
Prefabs
Projectiles
Assets > Prefabs > Projectiles
ClientProje...
ProjectileB...


Inspector
Console
ServerProjectile (Prefab Asset)
Open
Variant Parent
ProjectileBase

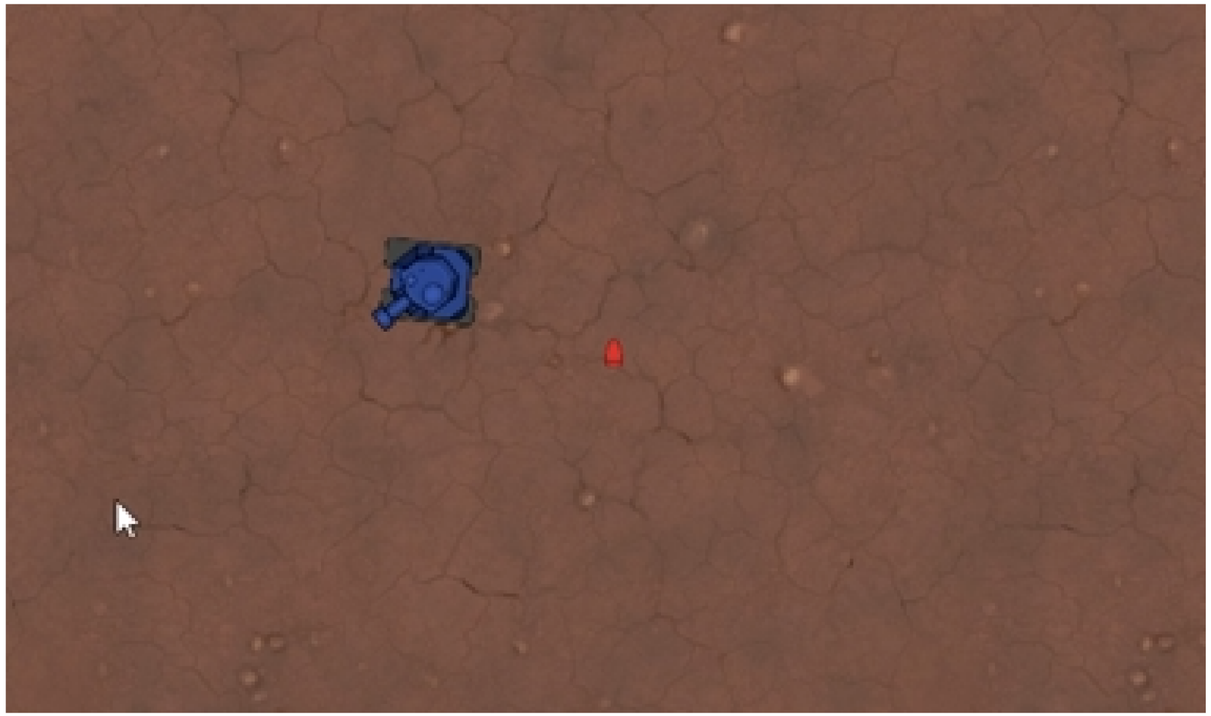

ClientProjectile
ClientProjectile
ShellSprite


Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1.1 Y 1.1 Z 1.1
Sprite Renderer
Sprite Tank_Shell
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprite-Lit-Default
Additional Settings
Sorting Layer Default
Order in Layer 0
Rendering Layer M Default
Sprite-Lit-Default (Material)
Shader Universal Render Pip
Edit...
Add Component

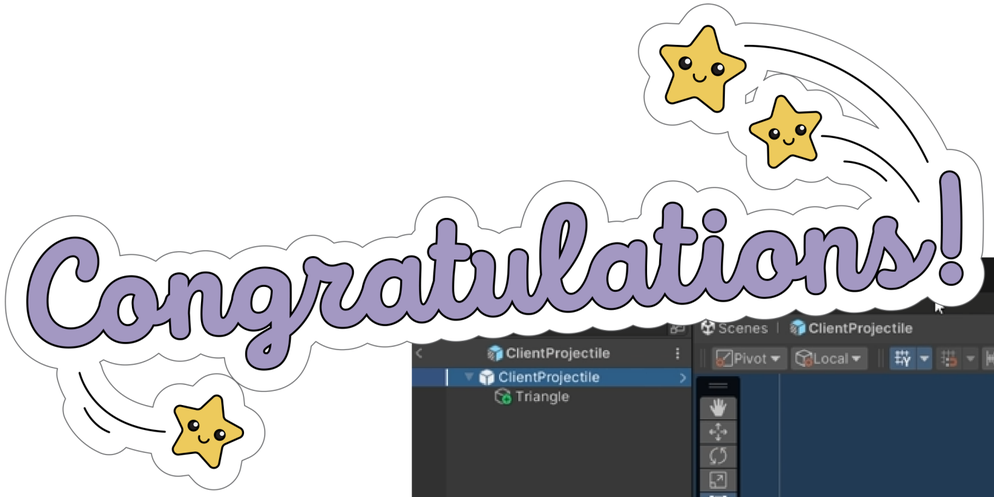
หากใครไม่เห็นลูกกระสุน ให้ Recheck ว่า

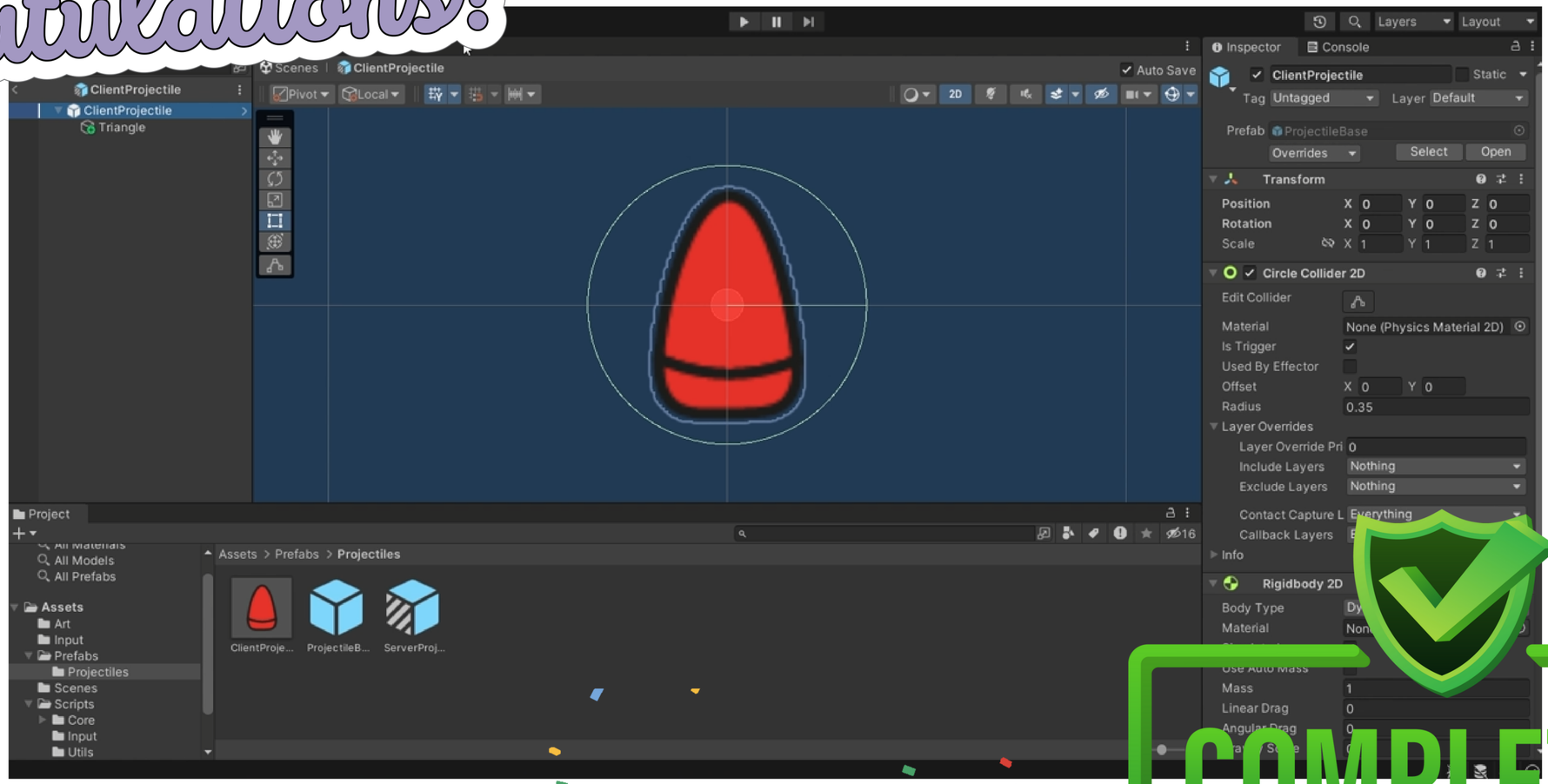


Time to Test “Networked Projectiles”

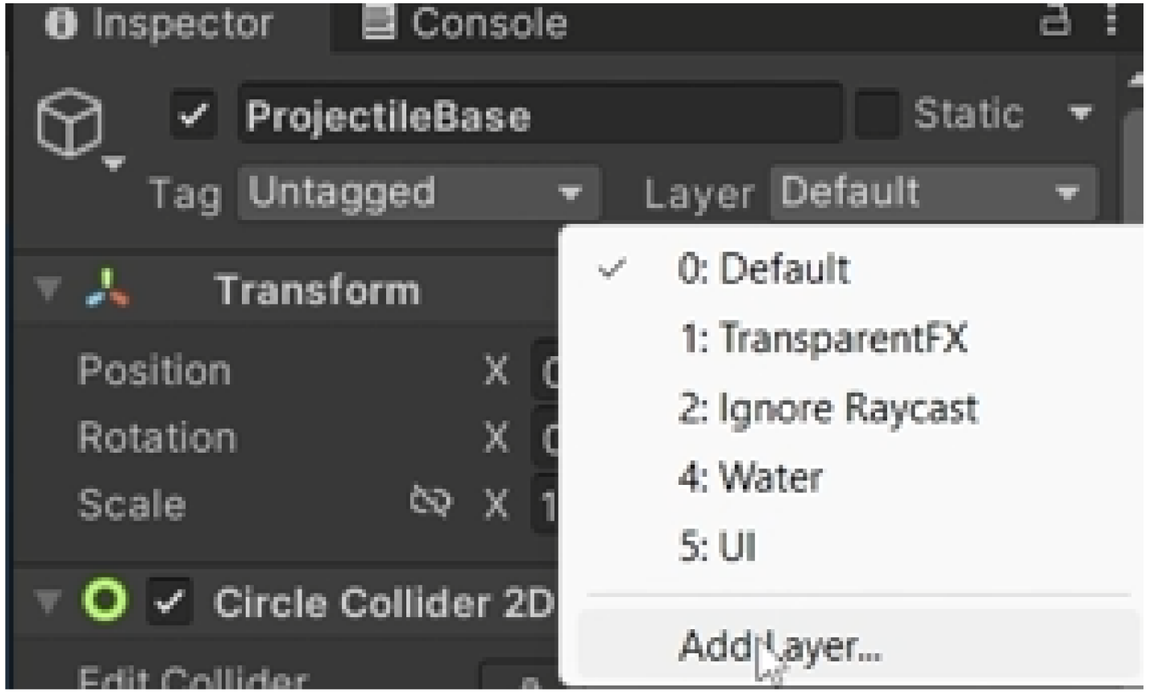


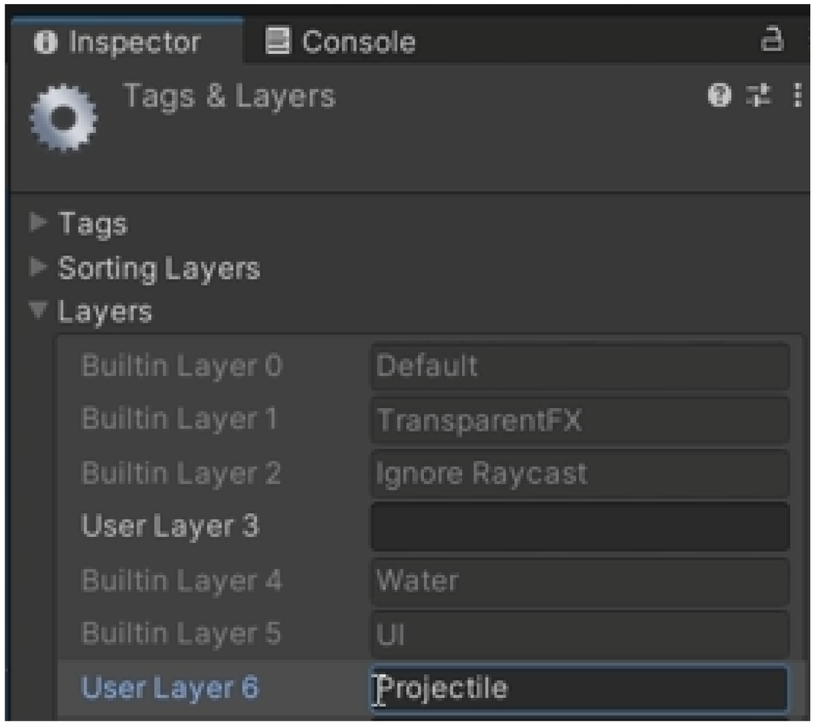
Networked Projectiles

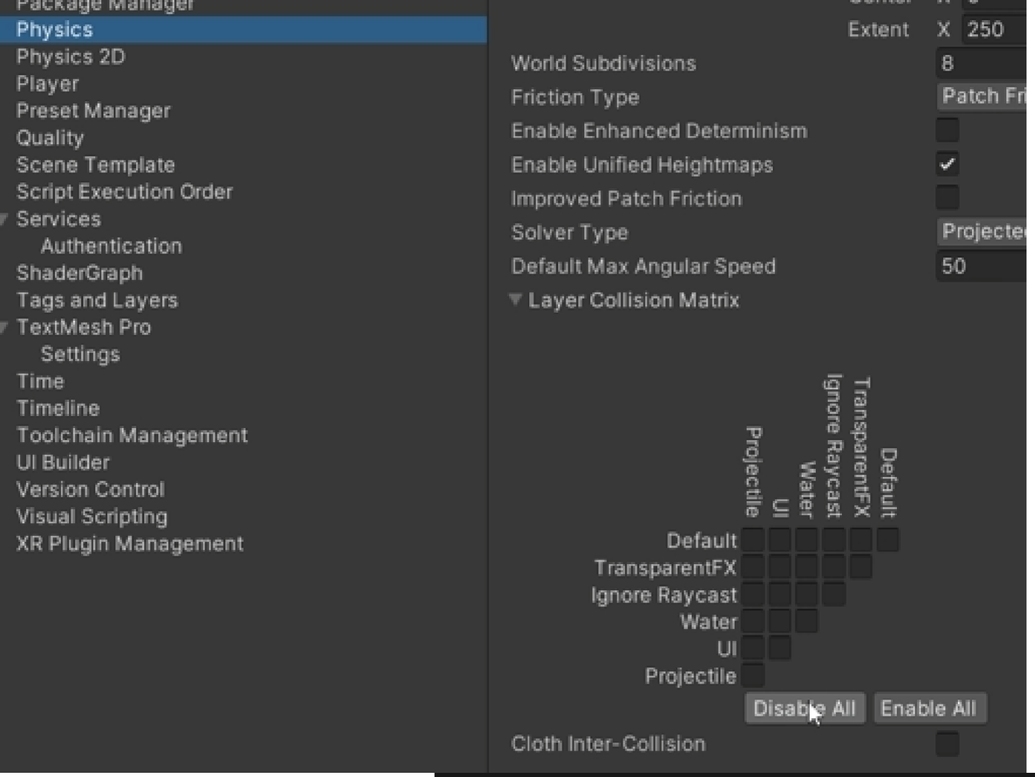


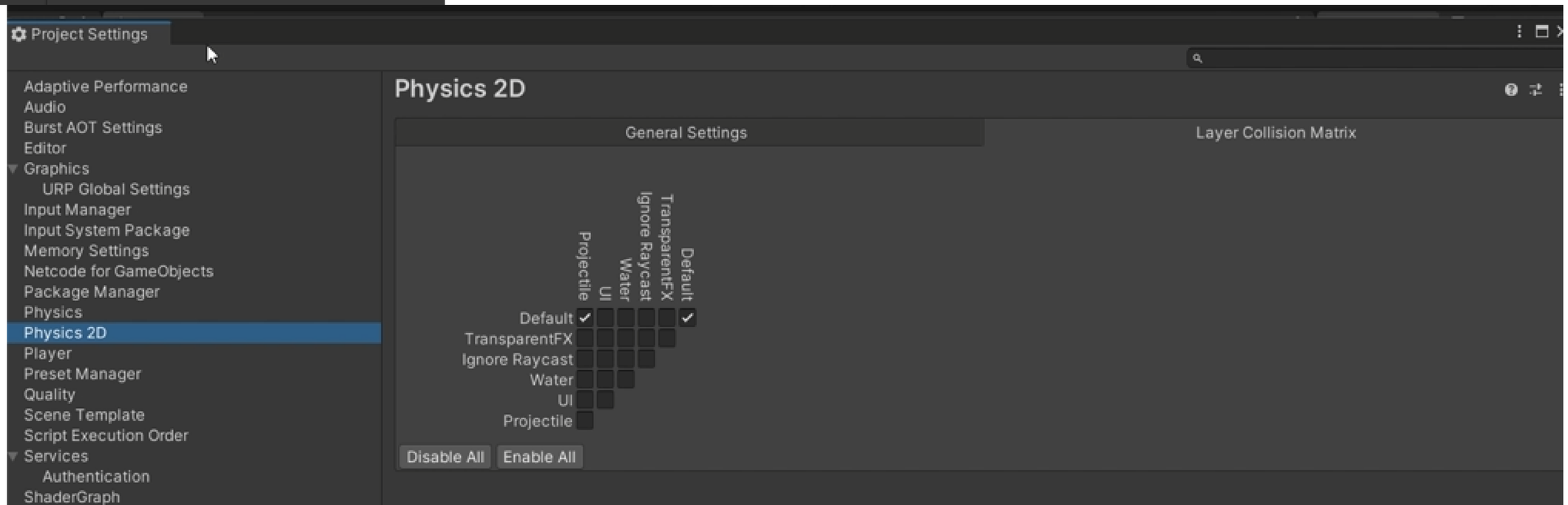


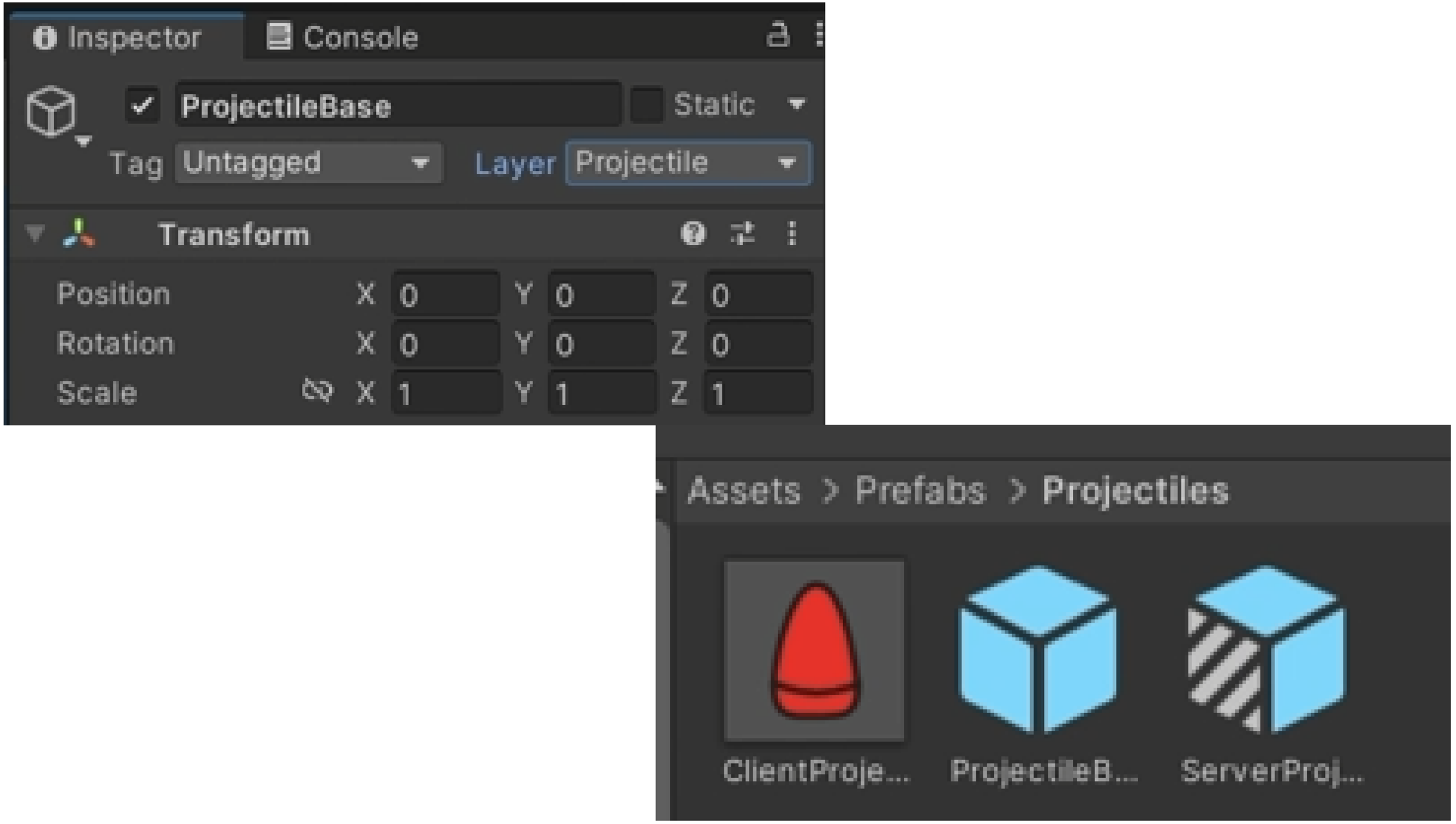
Firing Projectiles

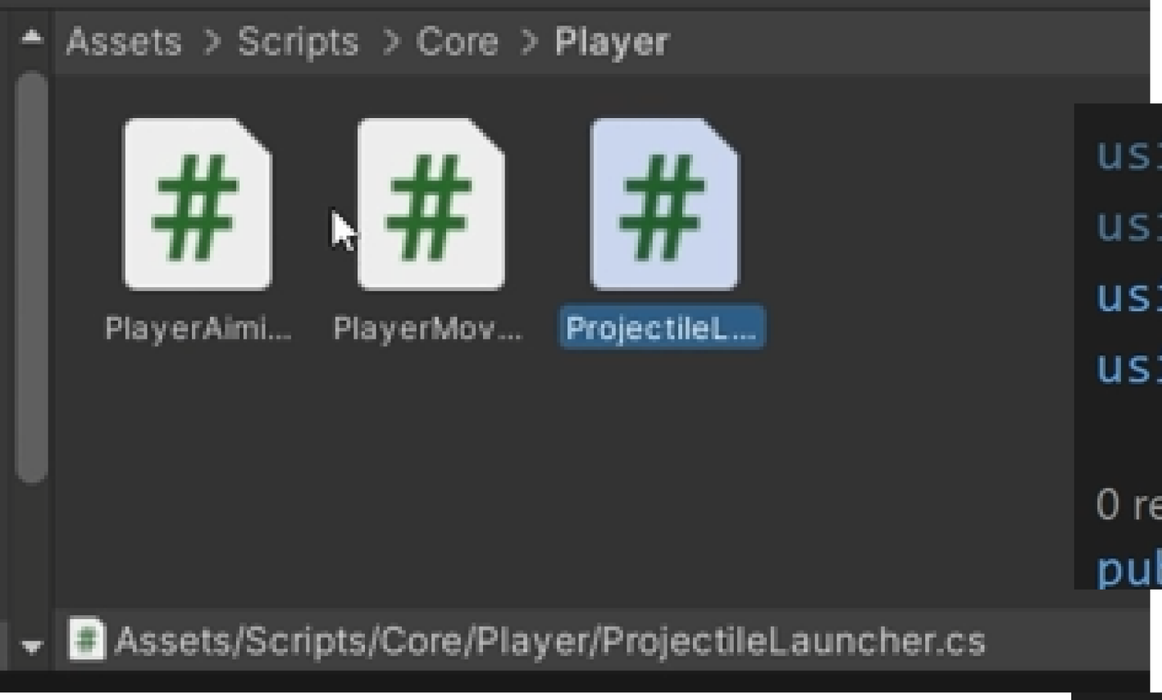

Inspector
Console
ProjectileBase
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Circle Collider 2D
0: Default
1: TransparentFX
2: Ignore Raycast
4: Water
5: UI
Add Layer...


Inspector
Console
Tags & Layers
Tags
Sorting Layers
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


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
Extent X 250
World Subdivisions 8
Friction Type
Enable Enhanced Determinism
Enable Unified Heightmaps
Improved Patch Friction
Solver Type
Default Max Angular Speed 50
Layer Collision Matrix
Default
TransparentFX
Ignore Raycast
Water
UI
Projectile
Disable All
Enable All
Cloth Inter-Collision

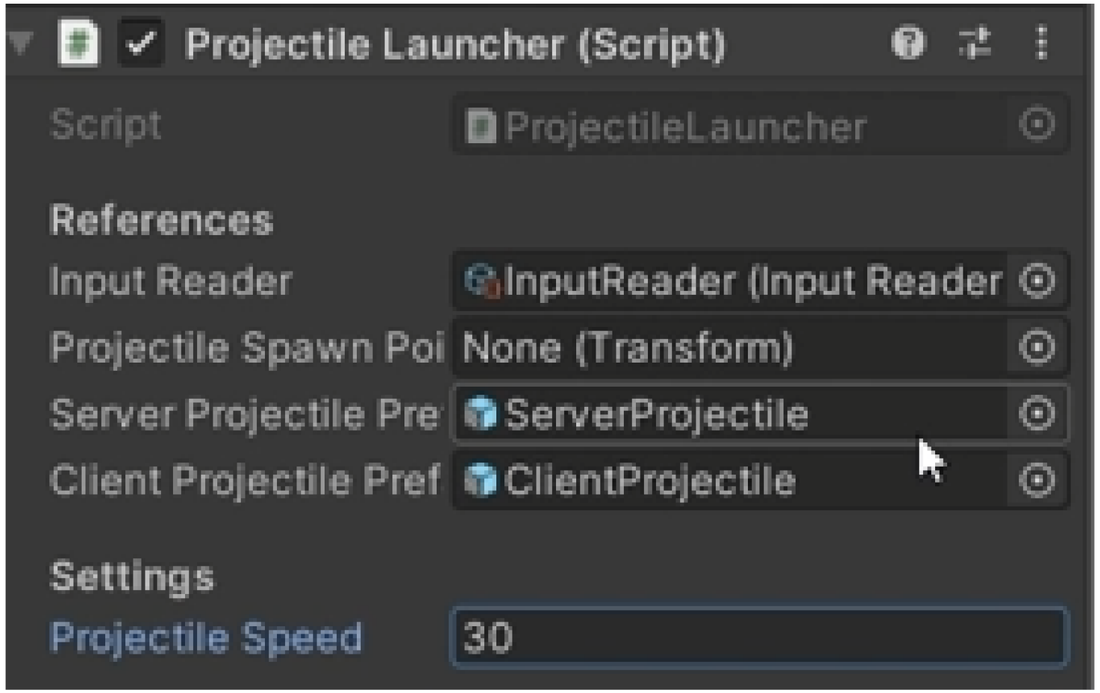

Project Settings
Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
URP Global Settings
Input Manager
Input System Package
Memory Settings
Netcode for GameObjects
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Authentication
ShaderGraph
Physics 2D
General Settings
Layer Collision Matrix
Default
TransparentFX
Ignore Raycast
Water
UI
Projectile
Disable All
Enable All

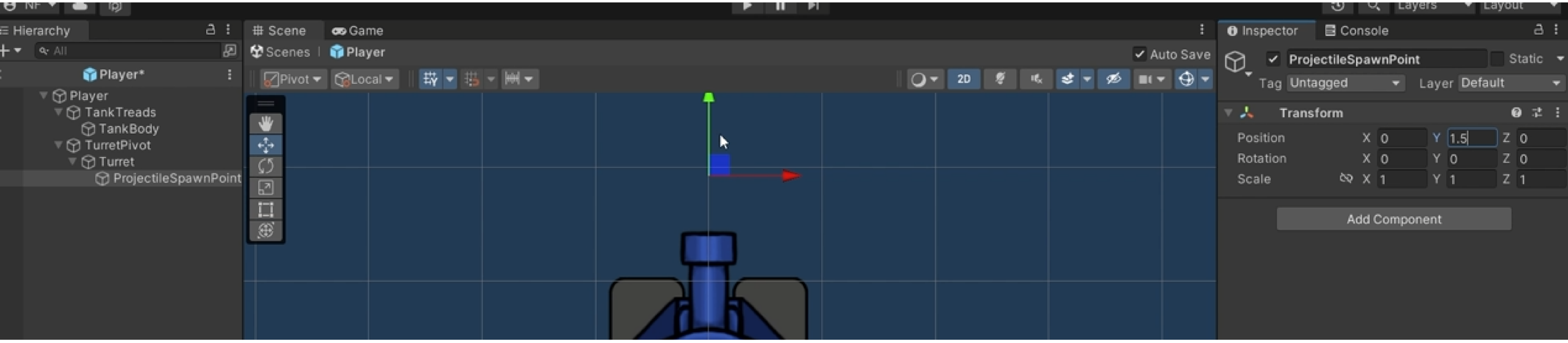

Inspector
Console
ProjectileBase
Static
Tag
Untagged
Layer
Projectile
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Assets > Prefabs > Projectiles
ClientProje...
ProjectileB...
ServerProj...

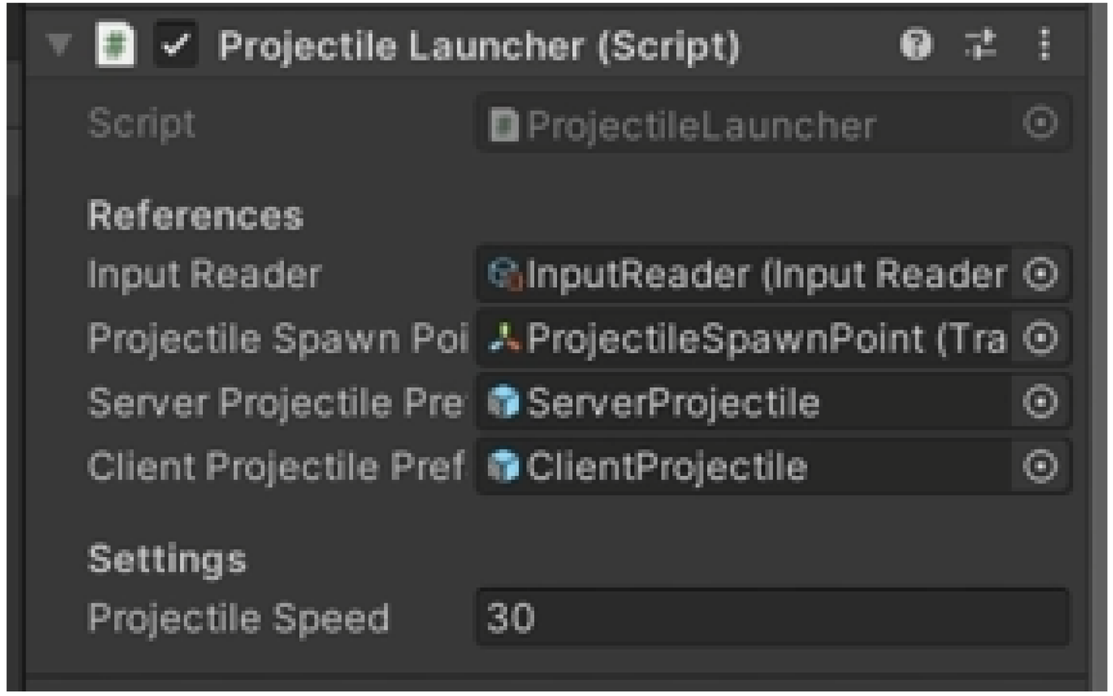



```
using System.Collections;
using System.Collections.Generic;
using Unity.Netcode;
using UnityEngine;

0 references
public class ProjectileLauncher : NetworkBehaviour
```

```
[Header("References")]
0 references
[SerializeField] private InputReader inputReader;
0 references
[SerializeField] private Transform projectileSpawnPoint;
0 references
[SerializeField] private GameObject serverProjectilePrefab;
0 references
[SerializeField] private GameObject clientProjectilePrefab;

[Header("Settings")]
0 references
[SerializeField] private float projectileSpeed;
```


Projectile Launcher (Script)
Script
ProjectileLauncher
References
Input Reader
InputReader (Input Reader
Projectile Spawn Poi
None (Transform)
Server Projectile Pre
ServerProjectile
Client Projectile Pref
ClientProjectile
Settings
Projectile Speed
30


Hierarchy
Player*
Player
TankTreads
TankBody
TurretPivot
Turret
ProjectileSpawnPoint
Scene
Game
Scenes
Player
Auto Save
Pivot
Local
2D
Inspector
Console
ProjectileSpawnPoint
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 1.5
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Add Component


Projectile Launcher (Script)
Script
ProjectileLauncher
References
Input Reader
InputReader (Input Reader
Projectile Spawn Poi
ProjectileSpawnPoint (Tra
Server Projectile Pre
ServerProjectile
Client Projectile Pref
ClientProjectile
Settings
Projectile Speed
30

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using Unity.Netcode;
4   using UnityEngine;
5
    Unity Script (1 asset reference) | 0 references
6   public class ProjectileLauncher : NetworkBehaviour
7   {
8       [SerializeField] private InputReader inputReade
9       [SerializeField] private Transform projectileSp
10      [SerializeField] private GameObject serverProje
11      [SerializeField] private GameObject clientProje
12
13      [Header("Settings")]
14      [SerializeField] private float projectileSpeed;
15
16      private bool shouldFire;
```

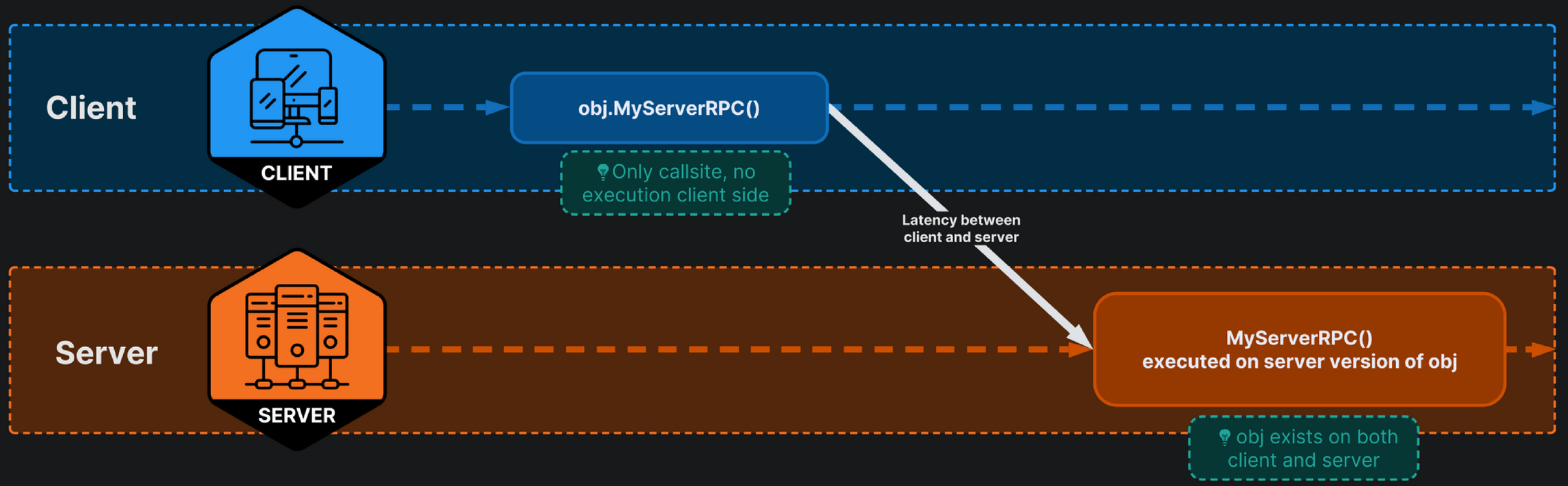
```
0 references
public override void OnNetworkSpawn()
{
    if (!IsOwner) {  return; }
    inputReader.PrimaryFireEvent += HandlePrimaryFire;
}

0 references
public override void OnNetworkDespawn()
{
    if (!IsOwner) { return; }
    inputReader.PrimaryFireEvent -= HandlePrimaryFire;
}
```

```
2 references
private void HandlePrimaryFire(bool shouldFire)
{
    this.shouldFire = shouldFire;
}
```

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

    projectileInstance.transform.up = direction;
}
```

```
0 references
private void Update()
{
    if (!IsOwner) { return; }

    if (!shouldFire) { return; }

    SpawnDummyProjectile(projectileSpawnPoint.position, projectileSpawnPoint.up);
}
```

Server RPCs



Client can invoke a server RPC on a Network Object. The RPC will be placed in the local queue and then sent to the server, where it will be executed on the server version of the same Network Object.

Sending Events with RPCs | Unity Multiplayer Networking

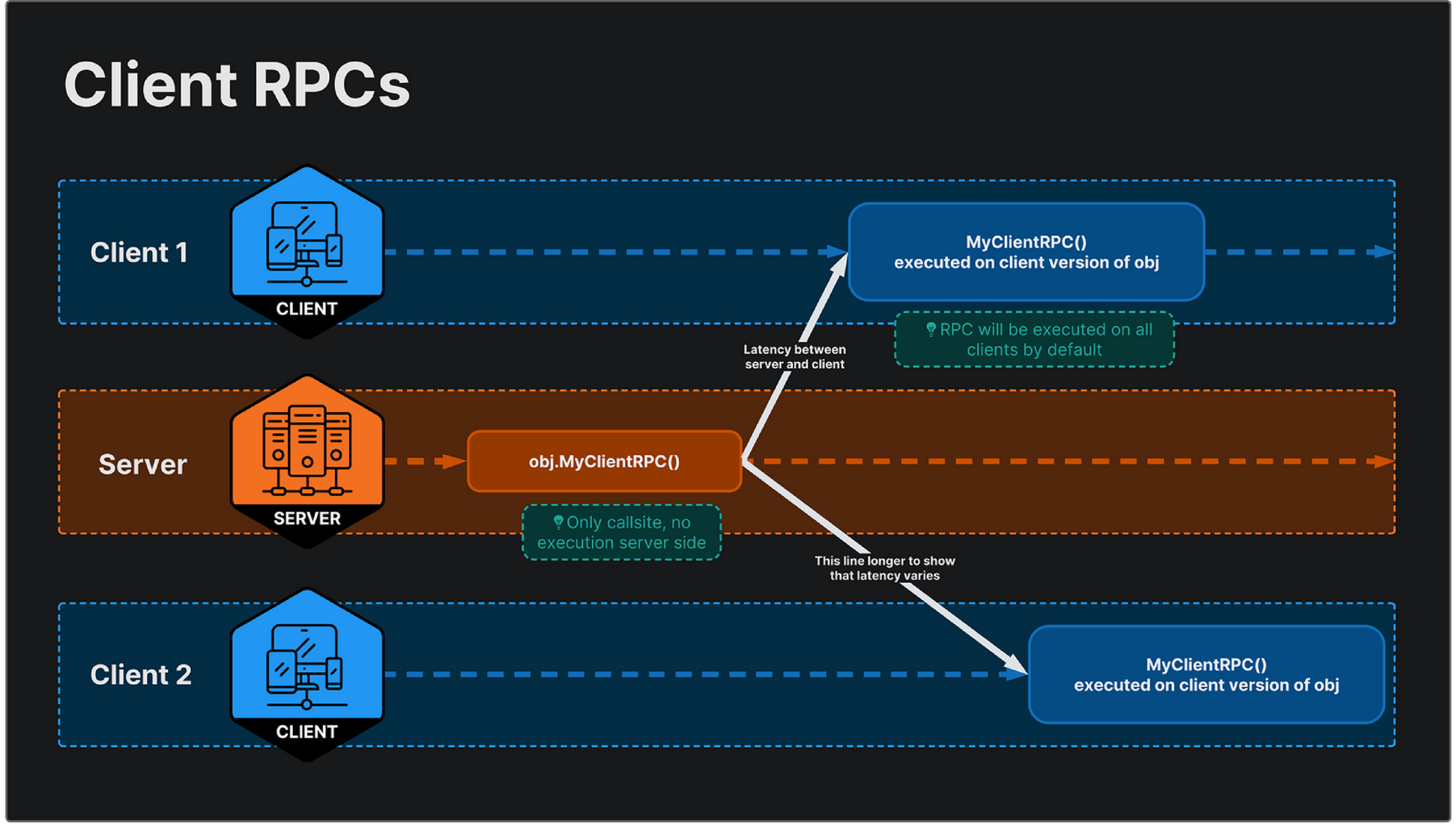
An introduction to the messaging system in Unity MLAPI, including RPC's and Custom Messages.

unity3d.com

```
[ServerRpc]
0 references
private void PrimaryFireServerRpc(Vector3 spawnPos, Vector3 direction)
{
    GameObject projectileInstance = Instantiate(
        serverProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;
}
```

Client Rpc

- **SpawnDummyProjectileClientRpc(Vector3, Vector3)**
- **[ClientRpc]** attribute
- Call from the server code
- Call **SpawnDummyProjectile** if we are not the owner



Server can invoke a client RPC on a Network Object. The RPC will be placed in the local queue and then sent to a selection of clients (by default this selection is "all clients"). When received by a client, RPC will be executed on the client's version of the same Network Object.

```
[ClientRpc]
1 reference
private void SpawnDummyProjectileClientRpc(Vector3 spawnPos, Vector3 direction)
{
    if(IsOwner) { return; }

    SpawnDummyProjectile(spawnPos,direction);
}
```

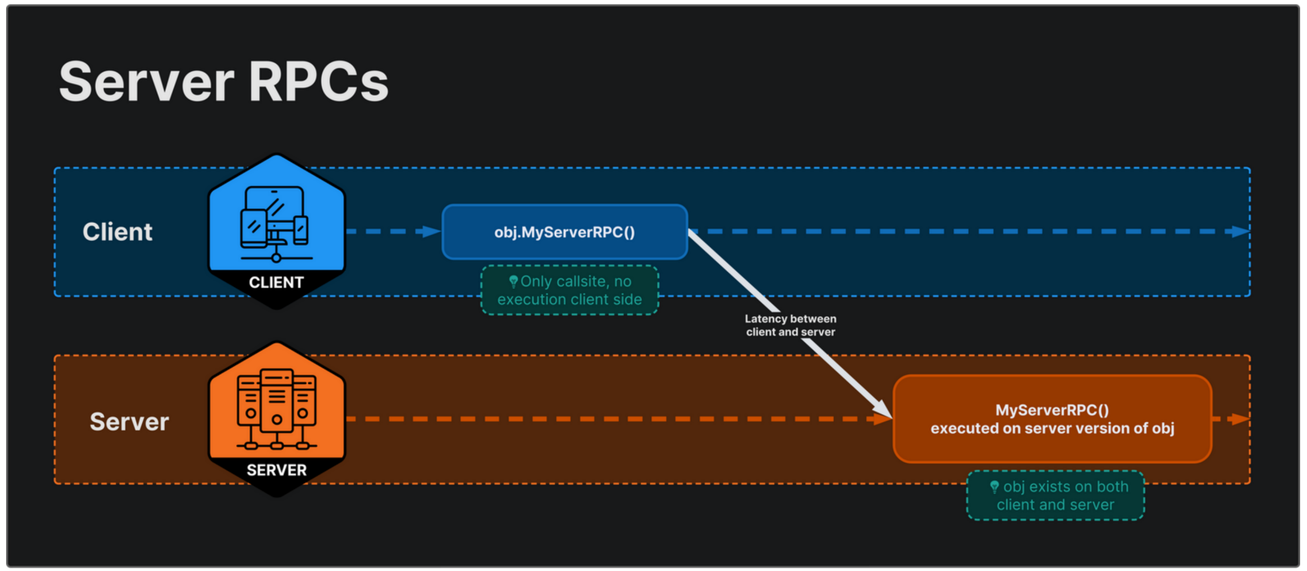
```
[ServerRpc]
1 reference
private void PrimaryFireServerRpc(Vector3 spawnPos, Vector3 direction)
{
    GameObject projectileInstance = Instantiate(
        serverProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;

    SpawnDummyProjectileClientRpc(spawnPos, direction);
}
```

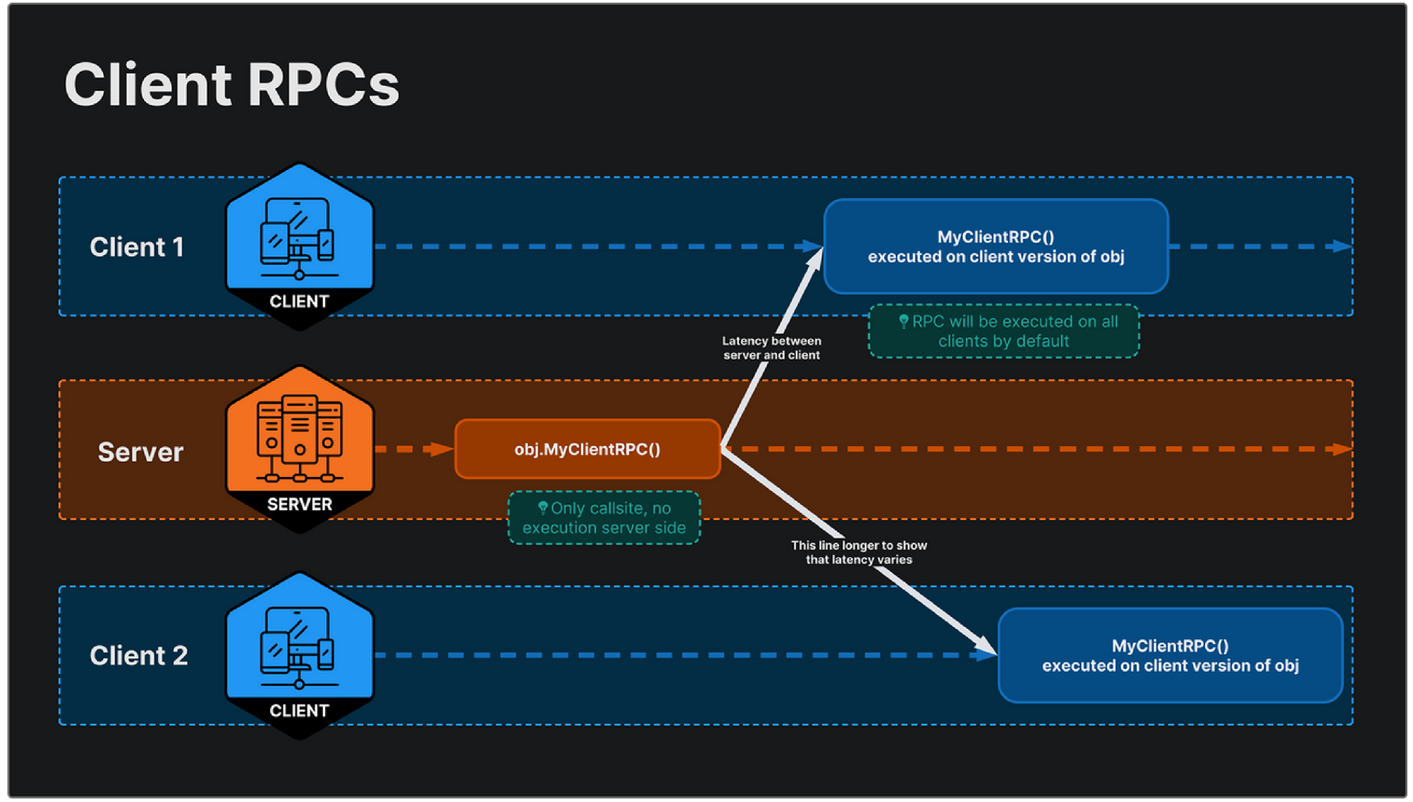
```
Unity Message | 0 references
void Update()
{
    if(!IsOwner) { return; }
    if(!shouldFire) { return; }

    PrimaryFireServerRpc(projectileSpawnPoint.position, projectileSpawnPoint.up);
    SpawnDummyProjectile(projectileSpawnPoint.position, projectileSpawnPoint.up);
}
```

ทบทวน

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

    projectileInstance.transform.up = direction;

    SpawnDummyProjectileClientRpc(spawnPos, direction);
}
[ClientRpc]
1 reference
private void SpawnDummyProjectileClientRpc(Vector3 spawnPos, Vector3 direction)
{
    if(IsOwner) { return; }

    SpawnDummyProjectile(spawnPos,direction);
}

2 references
private void SpawnDummyProjectile(Vector3 spawnPos, Vector3 direction)
{
    GameObject projectileInstance = Instantiate(
        clientProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;
}
```

```
Unity Message | 0 references
void Update()
{
    if(!IsOwner) { return; }
    if(!shouldFire) { return; }

    PrimaryFireServerRpc(projectileSpawnPoint.position, projectileSpawnPoint.up);
    SpawnDummyProjectile(projectileSpawnPoint.position, projectileSpawnPoint.up);
}
```



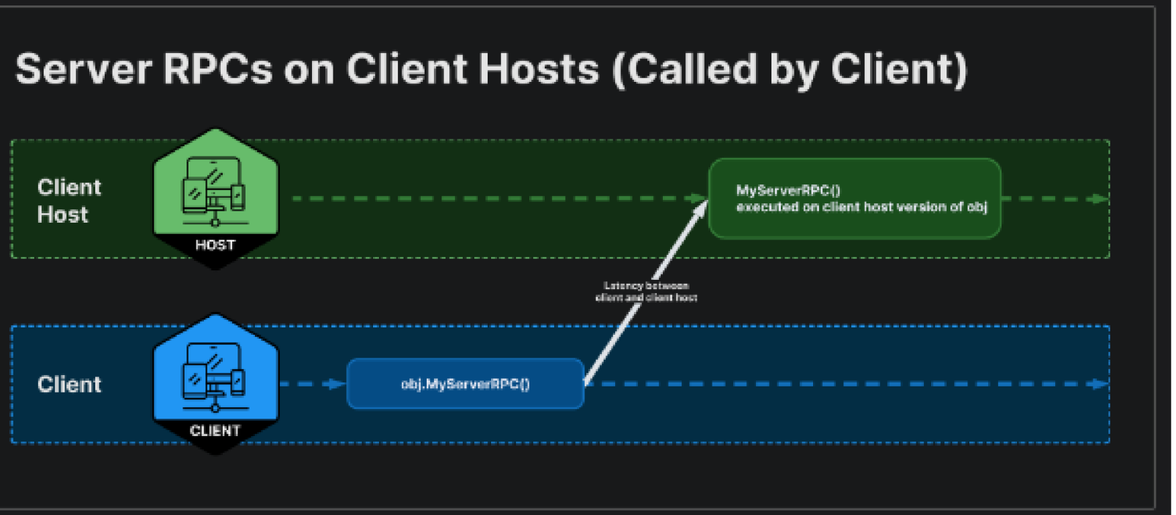


The following table details the execution of `ServerRpc` and `ClientRpc` functions:

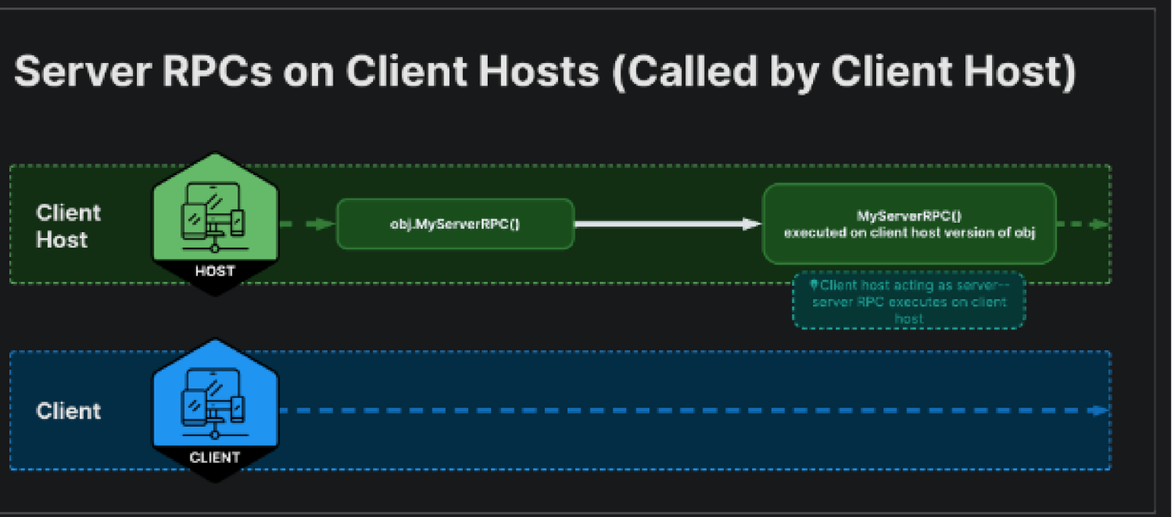
Function	Server	Client	Host (Server+Client)
ServerRpc Send	✗	✓	✓
ServerRpc Execute	✓	✗	✓
ClientRpc Send	✓	✗	✓
ClientRpc Execute	✗	✓	✓

An RPC function typically doesn't execute its body immediately since the function call is a stand-in for a network transmission. Since the host is both a client and a server, local RPCs targeting the host-server or host-client are invoked immediately. As such, avoid nesting RPCs when running in host mode as a ServerRpc method that invokes a ClientRpc method that invokes the same ServerRpc method (and repeat...) can cause a stack overflow.

กรณีที่เป็น Host



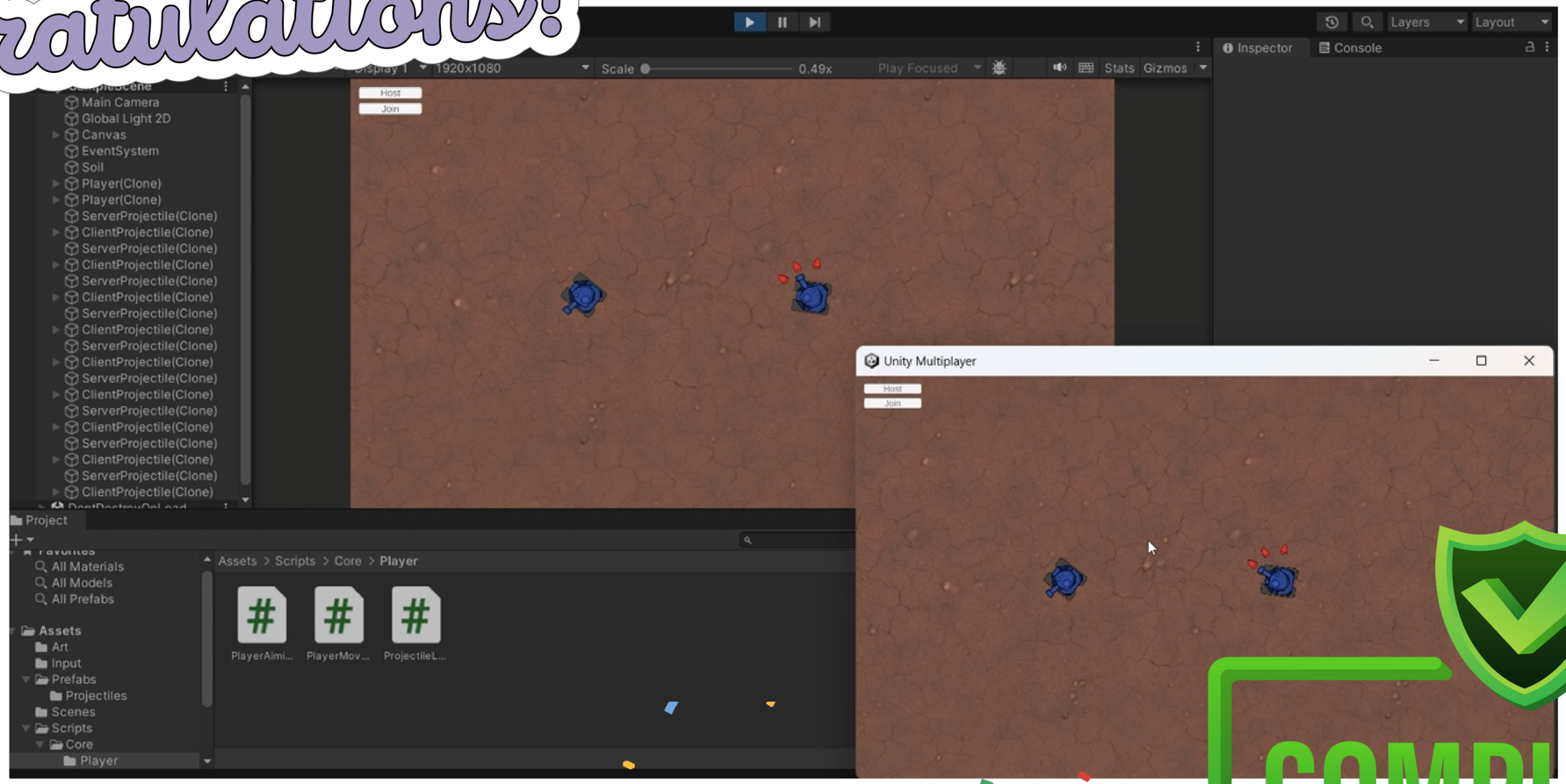
Clients can invoke server RPCs on Client Hosts the same way they can invoke server RPCs on the regular servers: the RPC will be placed in the local queue and then sent to the Client Host, where it will be executed on the Client Host's version of the same Network Object.



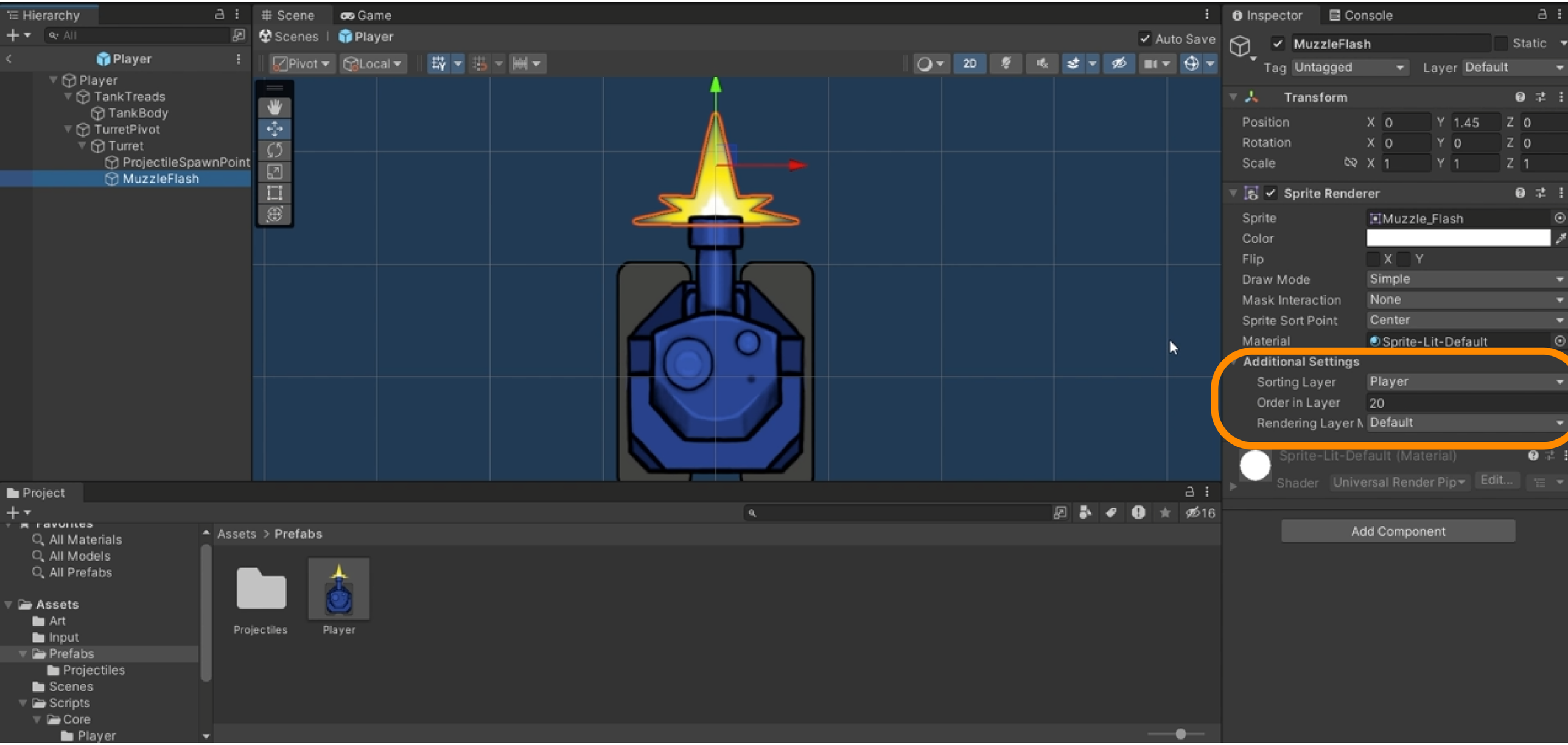
When a server RPC is invoked by the Client Host, the RPC will be placed in a local queue and then executed on the Client Host after a short delay. The same happens for pure servers.

Time to Test “Firing Projectiles”

Firing Projectiles

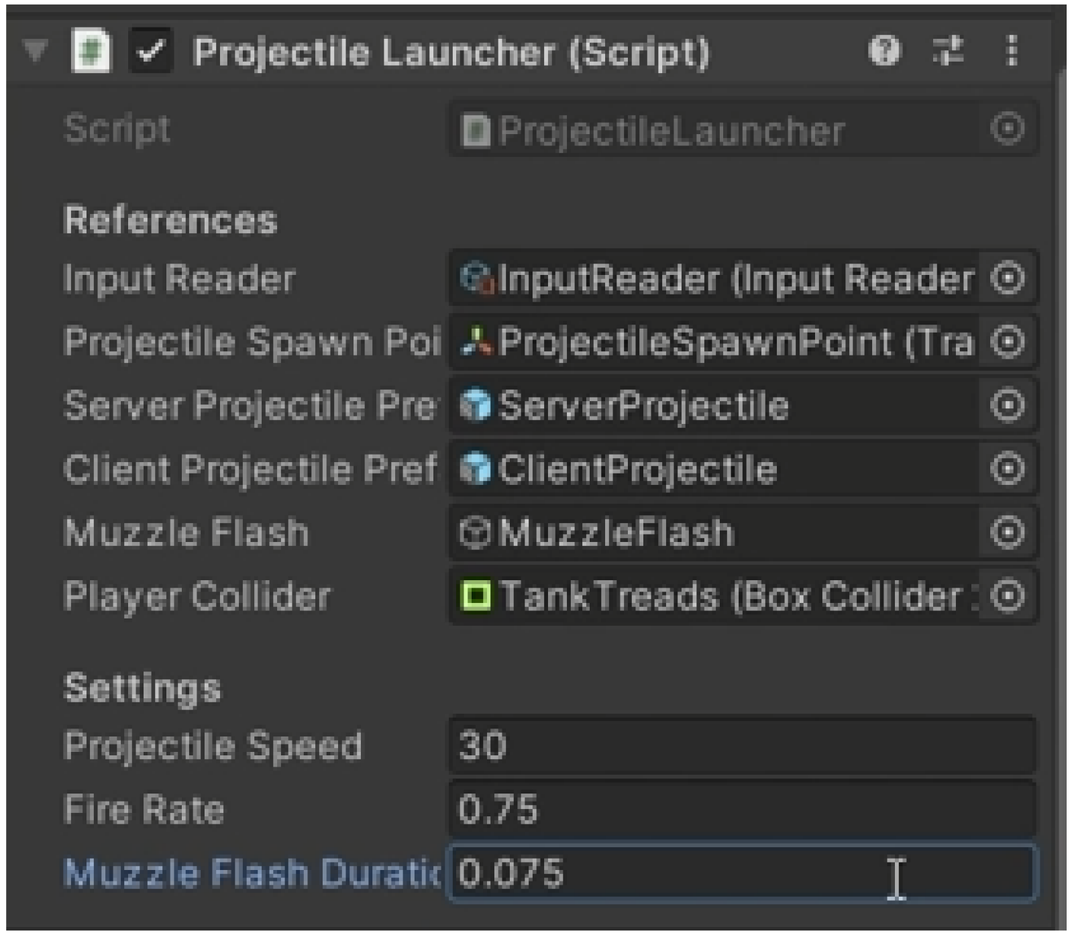


Firing Improvements


Hierarchy
Player
TankTreads
TankBody
TurretPivot
Turret
ProjectileSpawnPoint
MuzzleFlash
Scene
Game
Scenes
Player
Auto Save
Pivot
Local
2D
Inspector
Console
MuzzleFlash
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Sprite Renderer
Sprite
Muzzle_Flash
Color
Flip
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprite-Lit-Default
Additional Settings
Sorting Layer
Player
Order in Layer
20
Rendering Layer M
Default
Sprite-Lit-Default (Material)
Shader
Universal Render Pip
Edit...
Add Component
Project
Favorites
All Materials
All Models
All Prefabs
Assets
Art
Input
Prefabs
Projectiles
Scenes
Scripts
Core
Player
Assets > Prefabs
Projectiles
Player

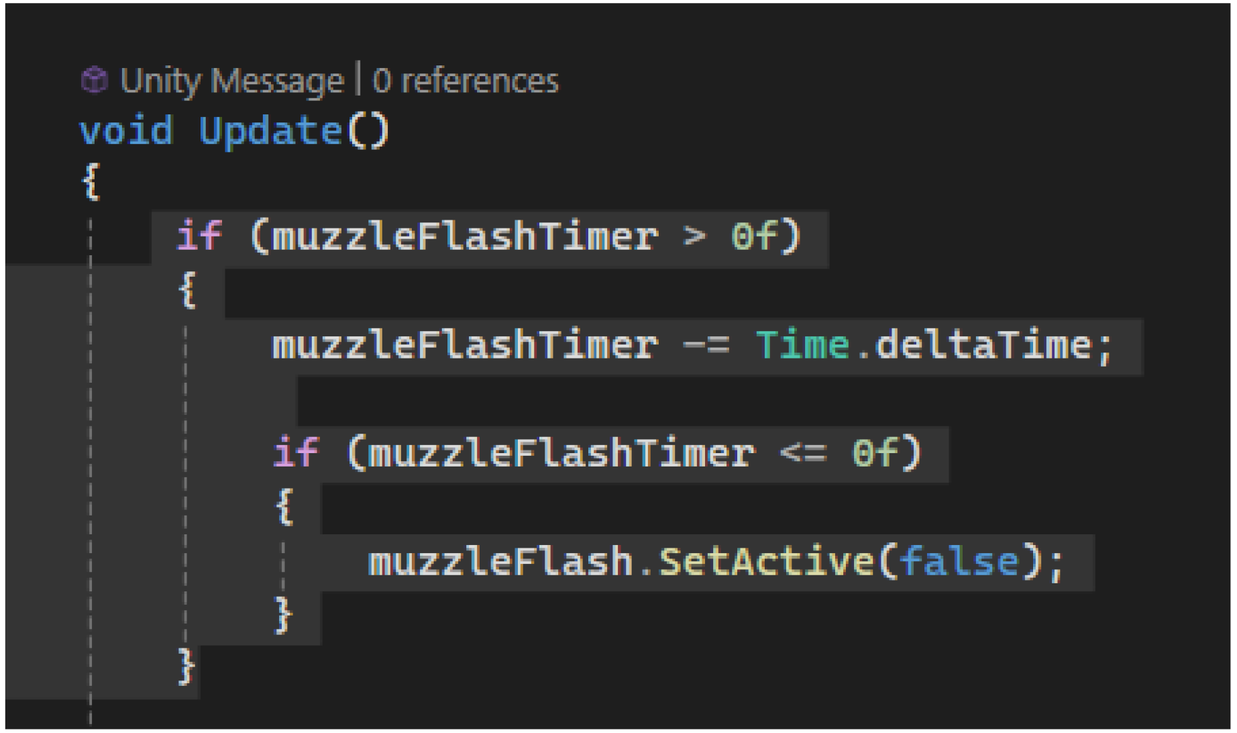
```
5
      Unity Script (1 asset reference) | 0 references
6     public class ProjectileLauncher : NetworkBehaviour
7     {
8         [SerializeField] private InputReader inputReader;
9         [SerializeField] private Transform projectileSpawnPoint;
10        [SerializeField] private GameObject serverProjectilePrefab;
11        [SerializeField] private GameObject clientProjectilePrefab;
12        [SerializeField] private GameObject muzzleFlash;
13        [SerializeField] private Collider2D playerCollider;
```

```
    [Header("Settings")]
    [SerializeField] private float projectileSpeed;
    [SerializeField] private float fireRate;
    [SerializeField] private float muzzleFlashDuration;
```


Projectile Launcher (Script)
Script
ProjectileLauncher
References
Input Reader
InputReader (Input Reader
Projectile Spawn Poi
ProjectileSpawnPoint (Tra
Server Projectile Pre
ServerProjectile
Client Projectile Pref
ClientProjectile
Muzzle Flash
MuzzleFlash
Player Collider
TankTreads (Box Collider
Settings
Projectile Speed
30
Fire Rate
0.75
Muzzle Flash Duratic
0.075

```
[Header("Settings")]
[SerializeField] private float projectileSpeed;
[SerializeField] private float fireRate;
[SerializeField] private float muzzleFlashDuration;
private bool shouldFire;
private float previousFireTime;
private float muzzleFlashTimer;
```

```
2 references
private void SpawnDummyProjectile(Vector3 spawnPos, Vector3 direction)
{
    muzzleFlash.SetActive(true);
    muzzleFlashTimer = muzzleFlashDuration;

    GameObject projectileInstance = Instantiate(
        clientProjectilePrefab,
        spawnPos,
        Quaternion.identity);
```


Unity Message | 0 references
void Update()
{
if (muzzleFlashTimer > 0f)
{
muzzleFlashTimer -= Time.deltaTime;
if (muzzleFlashTimer <= 0f)
{
muzzleFlash.SetActive(false);
}
}

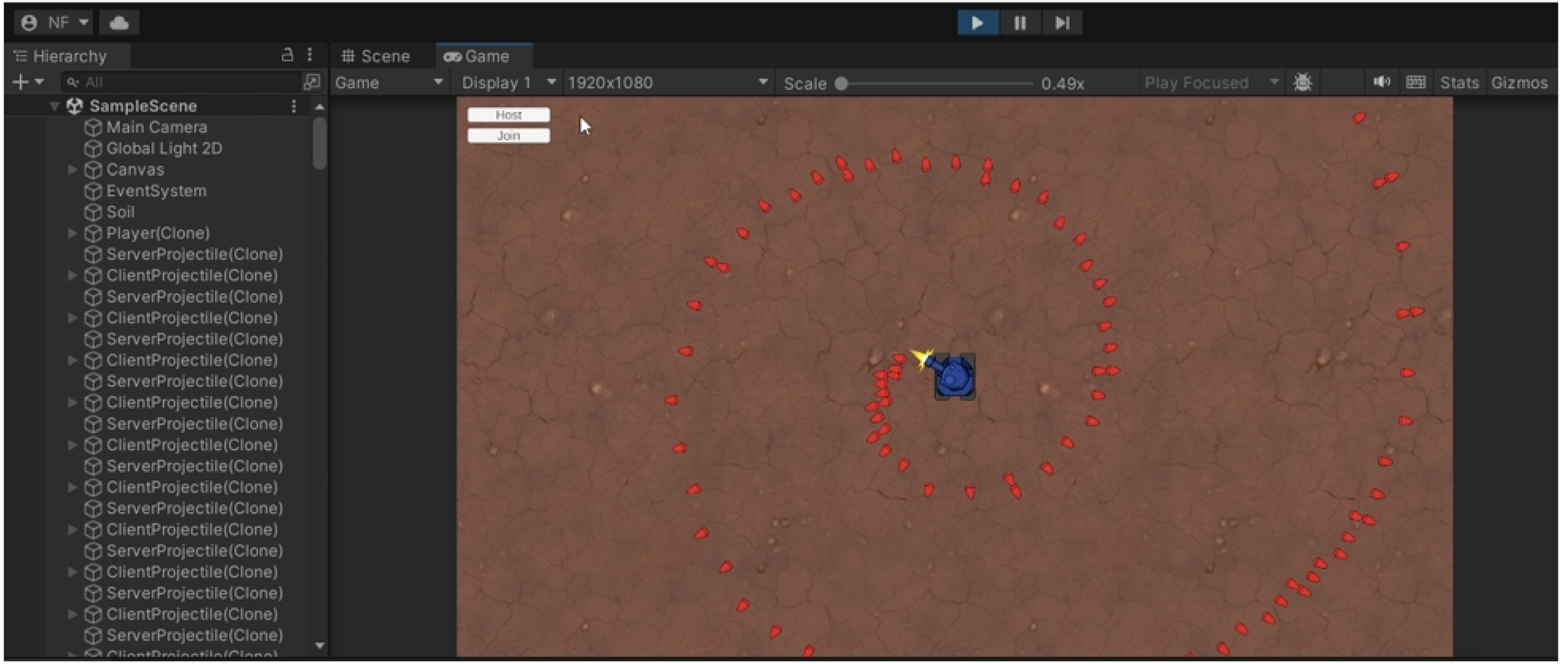
```
[ServerRpc]
1 reference
private void PrimaryFireServerRpc(Vector3 spawnPos, Vector3 direction)
{
    GameObject projectileInstance = Instantiate(
        serverProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;

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

```
2 references
private void SpawnDummyProjectile(Vector3 spawnPos, Vector3 direction)
{
    muzzleFlash.SetActive(true);
    muzzleFlashTimer = muzzleFlashDuration;

    GameObject projectileInstance = Instantiate(
        clientProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;

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

```
2 references
private void SpawnDummyProjectile(Vector3 spawnPos, Vector3 direction)
{
    muzzleFlash.SetActive(true);
    muzzleFlashTimer = muzzleFlashDuration;

    GameObject projectileInstance = Instantiate(
        clientProjectilePrefab,
        spawnPos,
        Quaternion.identity);

    projectileInstance.transform.up = direction;

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

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

```
[ServerRpc]
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<Rigidbody2D>(out Rigidbody2D rb))
    {
        rb.velocity = rb.transform.up * projectileSpeed;
    }

    SpawnDummyProjectileClientRpc(spawnPos, direction);
}
```

Time to Test “Physic Projectiles”

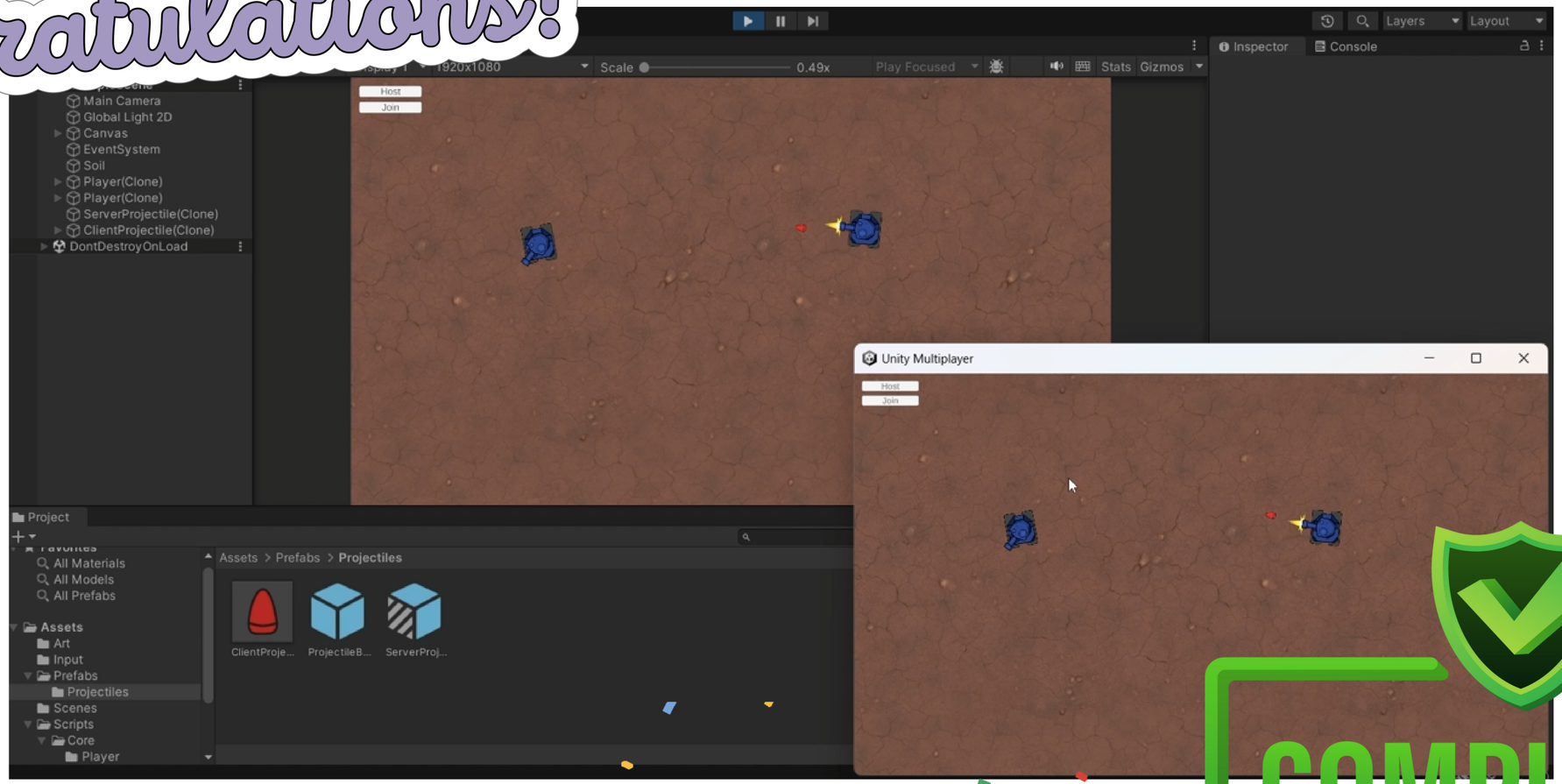


```
void Update()
{
    if (muzzleFlashTimer > 0f)
    {
        muzzleFlashTimer -= Time.deltaTime;

        if (muzzleFlashTimer <= 0f)
        {
            muzzleFlash.SetActive(false);
        }
    }

    if(!IsOwner) { return; }
    if(!shouldFire) { return; }

    if(Time.time < (1 / fireRate) + previousFireTime){ return; }

    PrimaryFireServerRpc(projectileSpawnPoint.position, projectileSpawnPoint.up);
    SpawnDummyProjectile(projectileSpawnPoint.position, projectileSpawnPoint.up);

    previousFireTime = Time.time;
}
```

หมายเหตุ* ในเวอร์ชั่นที่เราได้ทำไม่ได้ครอบคลุมเรื่องกันการโกงฝั่ง Client ซึ่งตามจริงเราอาจจะต้องใช้วิธีเรียก Check จาก Server ทั้งหมด แต่ต้องมาบริหารเรื่อง Lag และ Latency ในการส่ง/รับ ข้อมูลอยู่ดี

ดังนั้นขึ้นอยู่กับการออกแบบ ในตัวอย่างนี้จะเน้น Keep it simple และให้ฝั่ง Client เล่นได้ Smooth ที่สุดเป็นสำคัญเพื่อ Player Experience

Firing Improvements



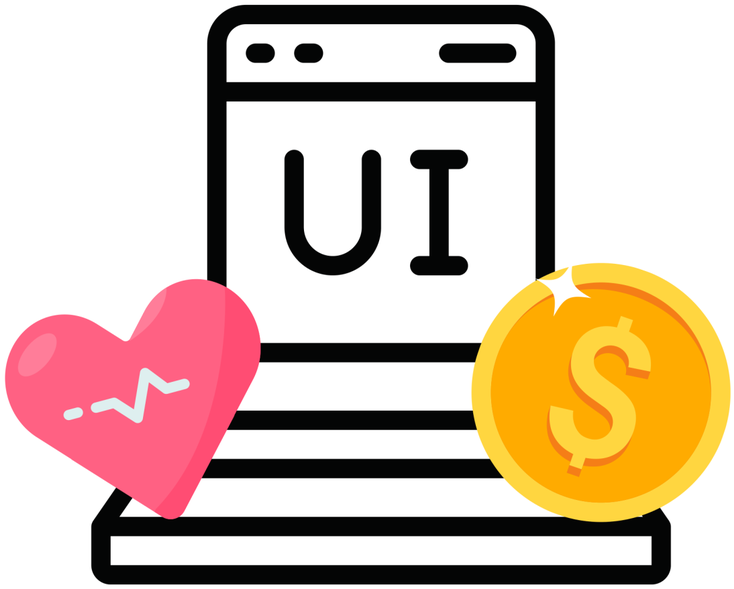
Assignment

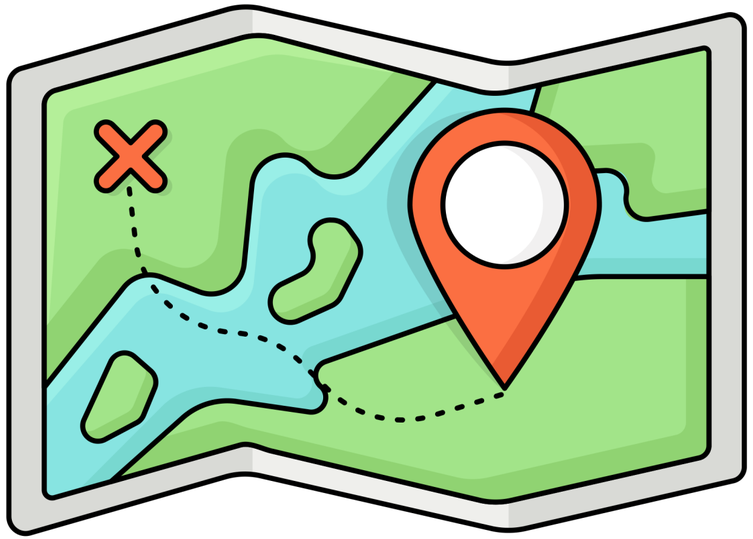
ให้ทำการถ่ายคลิปผลลัพธ์ของการทำ Workshop ตามเนื้อหา Workshop ในแต่ละหัวข้อนี้ พร้อมอธิบายประกอบ

- Player Movement
- Player Aiming
- Networked Projectiles
- Firing Projectiles
- Firing Improvements

NEXT WEEK

Sample Gameplay for Multiplayer 2





https://discord.gg/24xmUFHfzR

WOLVEDEN ACADEMY

THANK YOU

facebook.com/GameDevMew