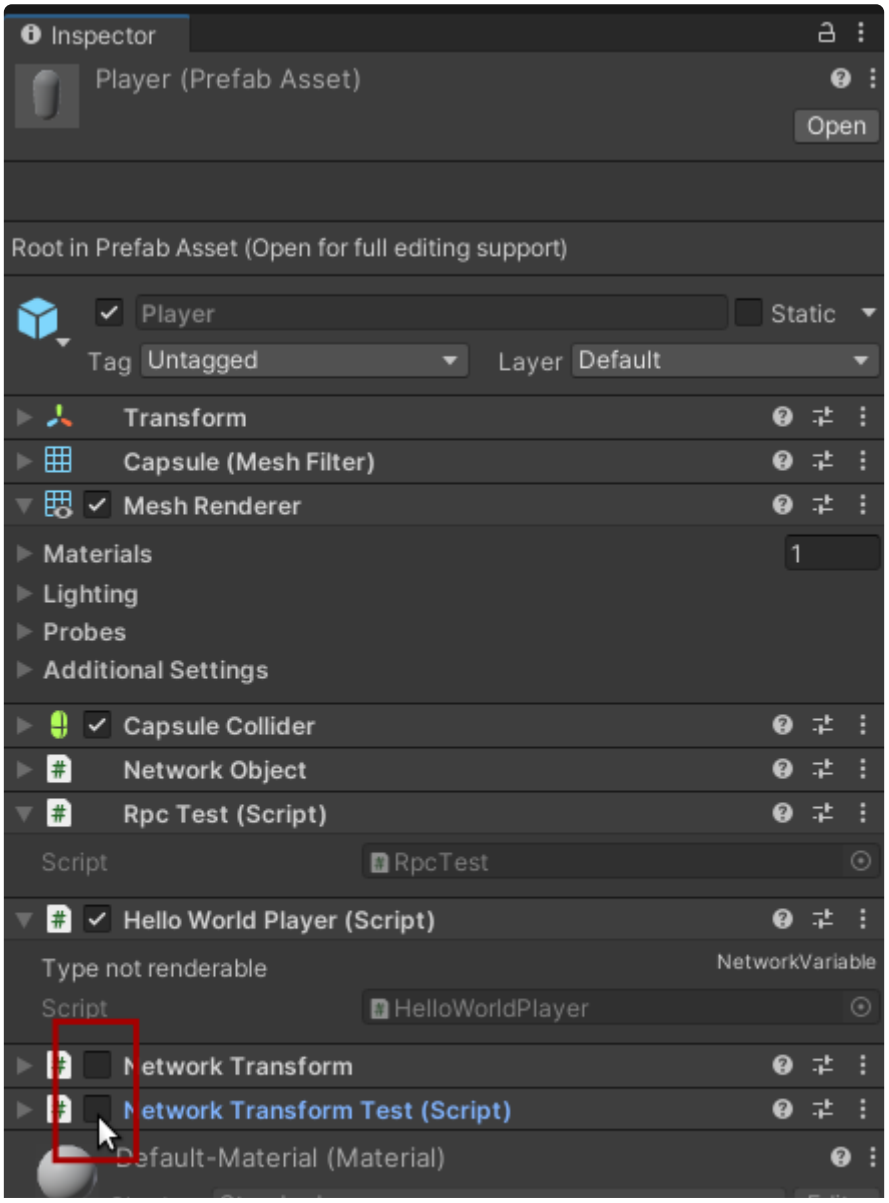
Práctica 2 - Movemento

Movemento multiplayer con variables de rede

Requisitos

1. Main project segundo o Manual de Unity: Get started with NGO
2. Inhabilitar o NetworkTransform para todas as instancias de *Player* para esta práctica.



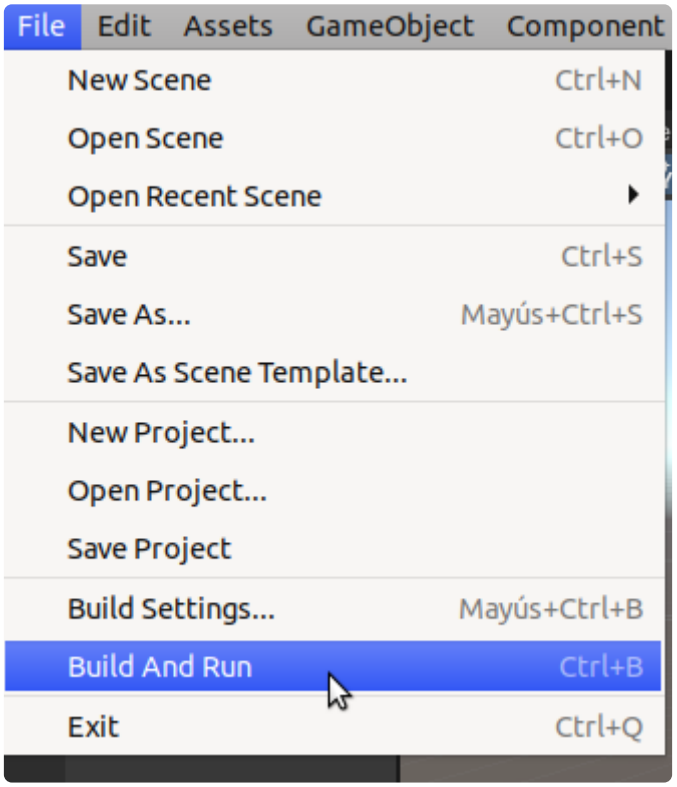
Obxectivo

Partindo do proxecto base NGO 'HelloWorld' da docu de Unity Multiplayer Networking, trátase de facer que as capsulas (Player gameObjects) respondan aos movementos esquerda, dereita, arriba e abaixo e de asegurarse de que cada movemento se reproduza en rede (que o player se mova en todos os equipos).

Esto débese conseguir sen usar o Network Transform para esta práctica, só variables Network e chamadas RPC

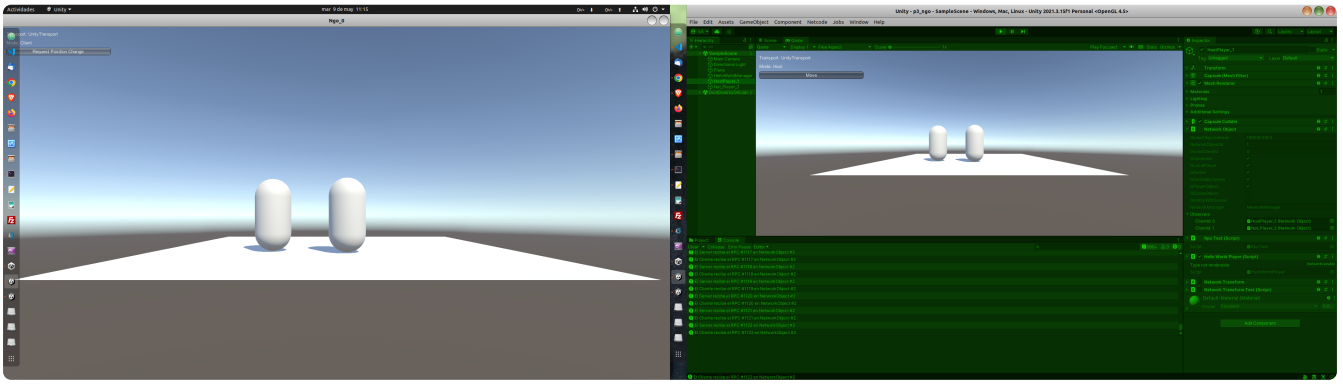
O escenario

Ao executar *Buid And Run*



Iníciase unha nova ventana de un equipo remoto virtual e así temos dous equipos.

Iniciamos en Unity o equipo Host para poder ver no equipo que fai de server as mensaxes por consola e no outro escollemos que sexa un ordenador cliente:

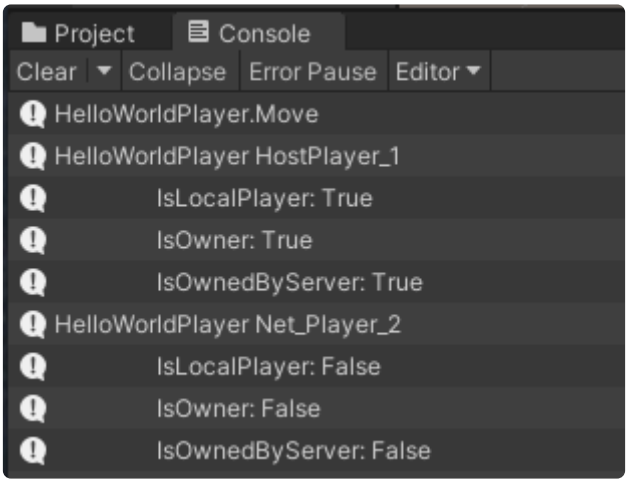


Aproveitando as variables de rede que se mostran no Inspector de cada Player, cambiarmos o player.name para distinguir na xerarquía o noso xogador cliente dos outros player en rede:

```
public class HelloWorldPlayer : NetworkBehaviour
{
        public NetworkVariable<Vector3> Position = new NetworkVariable<Vector3

        void Start()
        {
                var ngo = GetComponent<NetworkObject>();
                string uid = ngo.NetworkObjectId.ToString();
                // Dev
                if (ngo.IsOwnedByServer)
                {
                        gameObject.name = $"HostPlayer_{uid}";
                }
                else if (ngo.IsOwner)
                {
                        gameObject.name = $"Local_Player_{uid}";
                }
                else
                {
                        gameObject.name = "Net_Player_" + uid;
                }

                print($"HelloWorldPlayer {gameObject.name}");
                print($"\t IsLocalPlayer: {ngo.IsLocalPlayer}");
                print($"\t IsOwner: {ngo.IsOwner}");
                print($"\t IsOwnedByServer: {ngo.IsOwnedByServer}");
                // --- end Dev

        }

        public override void OnNetworkSpawn()
        {
                if (IsOwner)
                {
                        Move();
                }
        }

        ...
}
```

As instancias de player

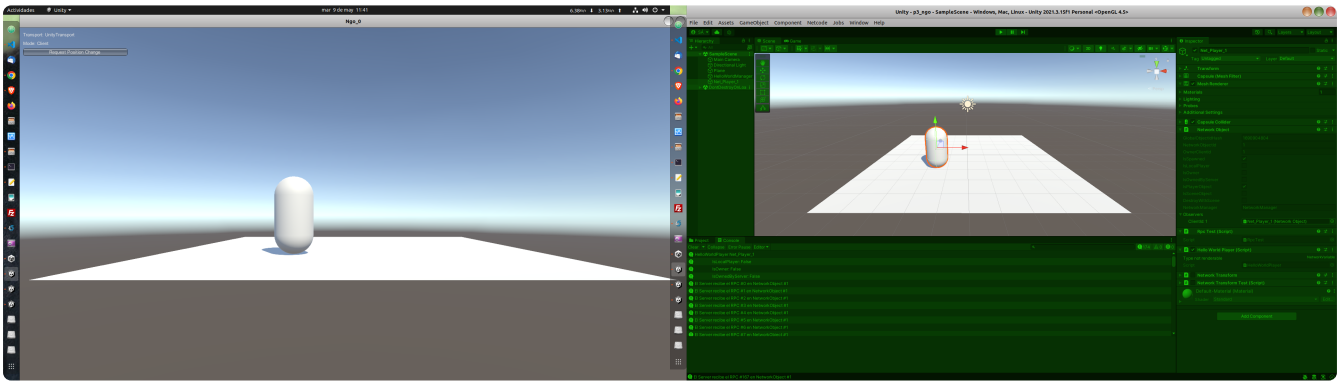
O código anterior mostra o seguinte resultado por consola:



É dicir, que temos as dúas instancias de Player no propio equipo, dado que podemos ver por consola a execución de este script en todos os xogadores: o xogador local do Host (HostPlayer_x) e o xogador en rede do outro equipo en rede(NetPlayer_y)

Cantas instancias de player hay en total?

Dúas por equipo coma se comprobou arriba, **en dous equipos 4**. Cada equipo instancia o seu propio xogador e tódolos contrincantes, excepto o server (Sólo server) que instancia tantas coma xogadores en rede hai que son todos menos sí mesmo:



Movemento á esquerda

Intento fallido

No update do script de player capturamos o Input para movernos á esquerda

HelloWordPlayer

```
void Update()
{
        if (Input.GetKeyDown(KeyCode.A) || Input.GetKeyDown(KeyCode.LeftArrow)
        {
                Debug.Log($"{gameObject.name}.HelloWorldPlayer.Update");
                Debug.Log($"\t Input A | Input left arrow");

                MoveToLeft();
        }


        // Asigna a posición da variable de rede ao game object
        transform.position = Position.Value;

        // Debug.Log($"{gameObject.name}.HelloWorldPlayer.Update");
        // Debug.Log($"\t transform.position: {transform.position}");
}
```

> A captura do imput de player para este exemplo debe ser GetKeyDown (unha única pulsación ate que se solte). En caso de usar GetAxis entenderase que o movimento persiste mentres se mantén a tecla pulsada, que son moitos frames por segundo;

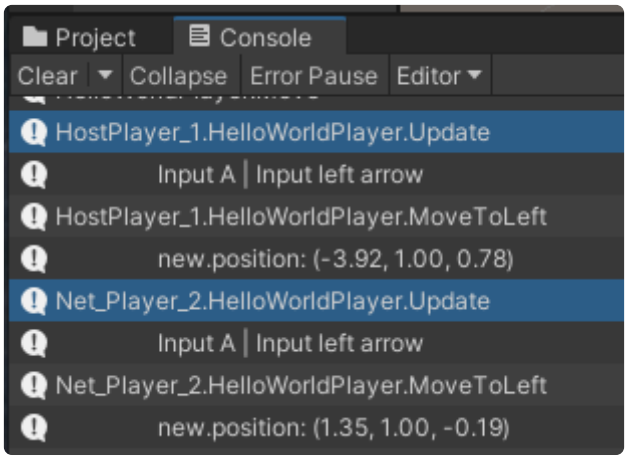
Copiamos e modificamos os métodos Move() e SubmitPositionRequestServerRpc() por MoveToLeft() e SubmitLeftPositionRequestServerRpc() respectivamente:

```
public void Move()
{
        if (NetworkManager.Singleton.IsServer)
        {
                var randomPosition = GetRandomPositionOnPlane();
                transform.position = randomPosition;
                Position.Value = randomPosition;

                Debug.Log($"HelloWorldPlayer.Move");
        }
        else
        {
                SubmitPositionRequestServerRpc();
        }
}

public void MoveToLeft()
{
        if (NetworkManager.Singleton.IsServer)
        {
                // Calculamos a posición que se pide por teclado
                Vector3 newPosition = transform.position + Vector3.left;
                // Actualizamos a posición na variable de rede
                Position.Value += Vector3.left;

                Debug.Log($"{gameObject.name}.HelloWorldPlayer.MoveToLeft");
                Debug.Log($"\t new.position: {newPosition}");
        }
        else
        {
                SubmitLeftPositionRequestServerRpc();
        }
}

[ServerRpc]
void SubmitLeftPositionRequestServerRpc(ServerRpcParams rpcParams = default)
{
        Position.Value += Vector3.left;

        Debug.Log($"{gameObject.name}.HelloWorldPlayer
                .SubmitLeftPositionRequestServerRpc");
        Debug.Log($"\t new.position: {Position.Value}");
}
```

```
[ServerRpc]
void SubmitPositionRequestServerRpc(ServerRpcParams rpcParams = default)
{
        Position.Value = GetRandomPositionOnPlane();
}
```

Pero NON FUNCIONA.

Cada vez que pulsamos á tecla esquerda móvense tódolos xogadores da escea:



Solución

Hai que agregar a condición de que, si quen pide mover por teclado non é o player propietario, non faga nada, en caso contrario, (quen pulsa a tecla de mover sí que é o xogador propietario) executamos o resto do código

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

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

        if (NetworkManager.Singleton.IsServer)
        {
                // Actualizamos a posición na variable de rede
                Position.Value += Vector3.left;

                Debug.Log($"{gameObject.name}.HelloWorldPlayer.MoveToLeft");
                Debug.Log($"\t new.position: {Position.Value}");
        }
        else
        {
                SubmitLeftPositionRequestServerRpc();
        }
}
```

Proposta de Solución

Pois nada, trasladamos a lóxica aos movementos arriba, dereita, abaixo e esquerda. Seguindo o sentido horario facemos un switch con 1, 2, 3 e 4 respectivamente