Lab 5

In this lab we are going to add a few more objects that hopefully will be useful for your game. We will add a simple scoring and UI.

# Scoring (UI and gameobject)

Apart from RPC, another way to sync in multiplayer in Unity is by using NetworkVariable. While both are considered as methods to transfer data and sync objects in network, RPC is normally used for transient events / information that only useful for a moment when it's received. On the other hand, NetworkVariables is for persistent states, which the information that will be around more than a moment (<https://docs-multiplayer.unity3d.com/netcode/current/learn/rpcvnetvar>).

In this lab example, we will use NetworkVariable to sync the score when the player knock on a box. This score will be sync for all clients including late joining clients.

1. Create a 2d square object (2D objects -> Sprites -> Square) and add RigidBody2D with Static type and a Box Collider 2D so this object can collide with the player

2. Assuming there could be many square objects you want to have later on, we will create a custom tag for the square objects. In this case lets name the custom tag to “coins”

3. To detect collision between the player and the coins and to update the score accordingly, we are going to update the movement codes (which if you follow the lab sheet your current active movement script is *movementPlayerRpc*. First we are going to create a Network Variable with a type of Integer called theScoreForEachPlayer. We are also going to give an initial value of 0 and most importantly we are going to set read permission to allow everybody (server and client) and write permission to Owner only.

private NetworkVariable<int> theScoreForEachPlayer2 = new NetworkVariable<int>(0, NetworkVariableReadPermission.Everyone, NetworkVariableWritePermission.Owner);

4. We are then going to check the collision so lets create an OnCollisionEnter2D to check if the player collides with the object with a tag “coins”.

5. If the player collide the Coins, only increase the networkvariable called theScoreForEachPlayer for the particular network object that collides with the mushroom is the owner. So add the code below to the OnCollissionEnter2D

void OnCollisionEnter2D(Collision2D target)

{

if (target.gameObject.tag.Equals("coins") == true)

{

if (!IsOwner) return;

theScoreForEachPlayer2.Value = theScoreForEachPlayer2.Value + 1;

}

}

6. When you look at the editor, you will be able to how many player objects in the game so to display the score on the GUI, we can just loop through the players object to get the network variable of it. First create an array of gameobjects at the start of the class

GameObject[] thePlayers2;

And then create the GUI

void OnGUI()

{

thePlayers2 = GameObject.FindGameObjectsWithTag("Player");

int x = 0;

foreach (GameObject respawn in thePlayers2)

{

GUI.Label(new Rect(10, 60 + (15 \* x), 300, 20), "PlayerID " + respawn.GetComponent<NetworkObject>().NetworkObjectId + " has the score of " + respawn.GetComponent<movementPlayerRPC>().theScoreForEachPlayer2.Value);

x++;

}

}

Please bear in mind, the accuracy of the sync is related to the latency in the network which can be seen that the score sometimes takes a split second to be updated. To manage the latency, you would need to make sure only essential data to be synced. Please note this method basically allows the network variable to be read by everyone and write by the Owner (is this safe?)