Lab 7

In this lab we are going to create a custom network variable which in this case would consist of the player owner id and score to keep the scoring. This is another way of creating the scoring and UI for the scoring.

1. In movementPlayerRPC, create a custom data type which in this case would use a struct that contains the playerID which is a ulong data and score which is an integer data. This data type would be inherited from INetworkSerizable interface and the implementation of the the interface should also included (click on the INetworkSerizable and choose implement interface). You would need to then specify what are the variables in the data type to be serialized

public struct MyPlayerData : INetworkSerializable

{

public ulong playerOwnderID;

public int score;

public void NetworkSerialize<T>(BufferSerializer<T> serializer) where T : IReaderWriter

{

serializer.SerializeValue(ref playerOwnderID);

serializer.SerializeValue(ref score);

}

}

1. Create a network variable as the custom type created earlier, assign default value and set the read and write permissions also to Everyone for Read and Owner for Write.

private NetworkVariable<MyPlayerData> myPlayerData = new NetworkVariable<MyPlayerData>(

new MyPlayerData

{

playerOwnderID = 0,

score = 0

}, NetworkVariableReadPermission.Everyone, NetworkVariableWritePermission.Owner

);

1. On the OnCollissionEnter2D, update the network variable when colliding with the coins

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

{

if (!IsOwner) return;

myPlayerData.Value = new MyPlayerData

{

playerOwnderIDStruct = OwnerClientId,

scoreStruct = myPlayerData.Value.scoreStruct + 1

};

}

1. In the OnGUI, replace the GUI Label to:

GUI.Label(new Rect(10, 60 + (15 \* x), 300, 20), "PlayerID " + respawn.GetComponent<Platformer2DUserControl>().myPlayerData.Value.playerOwnderID + " has the score of " + respawn.GetComponent<Platformer2DUserControl>().myPlayerData.Value.score + " using struct");

1. Additionally you can monitor the changes on the network variable by subscribing it to an event (you can put this in the NetworkSpawn())

public override void OnNetworkSpawn()

{

if (NetworkManager.Singleton.IsServer)

{

myPlayerData.OnValueChanged += (MyPlayerData previousValue, MyPlayerData newValue) =>

{

Debug.Log("OwnerClientId was " + previousValue.playerOwnderID + " now becomes " + newValue.playerOwnderID + " while score was " + previousValue.score + " now becomes " + newValue.score);

};

}

}