

For instantiation don't use - `Instantiate()` but use

`PhotonNetwork.Instantiate()`

# Add SpawnManager Object to map where you created the room and create an script SpawnManager

```
#region Photon Callback Methods
public override void OnJoinedRoom()
{
    if (PhotonNetwork.IsConnectedAndReady)
    {
        object playerSelectionNumber;
        if (PhotonNetwork.LocalPlayer.CustomProperties.TryGetValue(MultiplayerARSpinnerTopGame.PLAYER_SELECTION_NUMBER, out playerSelectionNumber))
        {
            Debug.Log("Player selection number is " + (int)playerSelectionNumber);
        }
    }
}
```

one check for good practice!

stores value store in var at PXQ it has to be there already to be assigned

getting custom properties values  
`TryGetValue(dictionaryRef.varname, out PXQ)`

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

public class SpawnManager : MonoBehaviourPunCallbacks
{
    public GameObject[] playerPrefabs;
    public Transform[] spawnPositions;

    // Start is called before the first frame update
    void Start()
    {
    }
}
```

SpawnManager

```

public override void OnJoinedRoom()
{
    if (PhotonNetwork.IsConnectedAndReady)
    {
        object playerSelectionNumber;
        if (PhotonNetwork.LocalPlayer.CustomProperties.TryGetValue(MultiplayerARSpinnerTopGame.PLAYER_SELECTION_NUMBER, out playerSele
        {
            Debug.Log("Player selection number is " + (int)playerSelectionNumber);

            int randomSpawnPoint = Random.Range(0, spawnPositions.Length-1);
            Vector3 instantiatePosition = spawnPositions[randomSpawnPoint].position;

            PhotonNetwork.Instantiate(playerPrefabs[(int)playerSelectionNumber].name, instantiatePosition, Quaternion.identity);
            GameObject PhotonNetwork.Instantiate(string prefabName, Vector3 position, Quaternion rotation, [byte group = 0], [object[] data = null])
        }
    }
}

```

you need to **PhotonView** Component to  
your **beyblade** PREFABS

# PhotonView Component

First and foremost, we need to have a PhotonView component attached to our prefab. A PhotonView is what connects together the various instances on each computers, and define what components to observe and how to observe these components.

1. Add a PhotonView Component to `My Robot Kyle`
2. Set the `Observe Option` to `Unreliable On Change`
3. Notice `PhotonView` warns you that you need to observe something for this to have any effects

Let's set up what we are going to observe, and then we'll get back to this `PhotonView` component and finish its setup.

[Back To Top](#)

