

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

```
public class CreateAndJoinRooms : MonoBehaviourPunCallbacks
```

```
{
    public InputField crearInput;
    public GameObject panelLobby, panelSala;
    public Text nombreSala;
    public Salaltem salaltemPrefab;
    List<Salaltem> salaltemLista = new List<Salaltem>();
    public Transform objetoContenido;
    private List<ItemJugador> listaItemsJugadores = new List<ItemJugador>();
    public ItemJugador itemJugadorPrefab;
    public Transform itemJugadorParent;

    public GameObject botonJugar;

    private void Start()
    {
        PhotonNetwork.JoinLobby();
        if (PhotonNetwork.CurrentRoom != null)
        {
            PhotonNetwork.LeaveRoom();
        }
    }

    public void CrearSala() {
        if (crearInput.text.Length > 0)
        {
            PhotonNetwork.CreateRoom(crearInput.text, new RoomOptions() { MaxPlayers = 6 , BroadcastPropsChangeToAll =
true});
        }
    }

    public override void OnJoinedRoom()
    {
        panelLobby.SetActive(false);
        panelSala.SetActive(true);
        nombreSala.text = " " + PhotonNetwork.CurrentRoom.Name;
        UpdateListaJugadores();
    }

    public void UnirSala(string nombreSala) {
        PhotonNetwork.JoinRoom(nombreSala);
    }

    public override void OnRoomListUpdate(List<RoomInfo> roomList)
    {
        UpdateRoomList(roomList);
    }

    private void UpdateRoomList(List<RoomInfo> listaSalas)
    {
        foreach (Salaltem salaltem in salaltemLista) {
            Destroy(salaltem.gameObject);
            Destroy(salaltem);
        }
    }
}
```

```
salaltemLista.Clear();
```

```
foreach (RoomInfo salaInfo in listaSalas)
```

```
{  
    if (salaInfo.PlayerCount != 0)  
    {  
        Salaltem nuevaSala = Instantiate(salaltemPrefab, objetoContenido);  
        nuevaSala.SetNombreSala(salaInfo.Name);  
        salaltemLista.Add(nuevaSala);  
    }  
}
```

```
public void OnClickSalirSala() {  
    PhotonNetwork.LeaveRoom();  
}
```

```
public override void OnLeftRoom()  
{  
    panelSala.SetActive(false);  
    panelLobby.SetActive(true);  
}
```

```
public override void OnConnectedToMaster()  
{  
    PhotonNetwork.JoinLobby();  
}
```

```
void UpdateListaJugadores() {  
    foreach(ItemJugador item in listaltemsJugadores) {  
        Destroy(item.gameObject);  
        Destroy(item);  
    }  
    listaltemsJugadores.Clear();  
}
```

```
if (PhotonNetwork.CurrentRoom == null) {  
    return;  
}
```

```
foreach (KeyValuePair<int, Player> player in PhotonNetwork.CurrentRoom.Players) {  
    ItemJugador nuevotemJugador = Instantiate(itemJugadorPrefab, itemJugadorParent);  
    nuevotemJugador.SetPlayerInfo(player.Value);  
    if (player.Value == PhotonNetwork.LocalPlayer) {  
        nuevotemJugador.AplicarCambiosLocales();  
    }  
    listaltemsJugadores.Add(nuevoltemJugador);  
}
```

```
public override void OnPlayerEnteredRoom(Player newPlayer)  
{  
    UpdateListaJugadores();  
}
```

```
public override void OnPlayerLeftRoom(Player otherPlayer)  
{  
    UpdateListaJugadores();  
}
```

```
private void Update()  
{
```

```
    if (PhotonNetwork.IsMasterClient && PhotonNetwork.CurrentRoom.PlayerCount >0)//CAMBIAR EL 0 POR 1 CUANDO NO  
    ESTÁ% EN PRUEBA EL JUEGO  
    {  
        botonJugar.SetActive(true);  
    }
```

```
}
else {
    botonJugar.SetActive(false);
}
}

public void OnClickBotonJugar() {
    ReparteSpawns();
    PhotonNetwork.LoadLevel("Ovalo");
}

private void ReparteSpawns()
{
    ExitGames.Client.Photon.Hashtable propiedadesAUX = new ExitGames.Client.Photon.Hashtable();
    int i = 0;
    foreach (KeyValuePair<int, Player> player in PhotonNetwork.CurrentRoom.Players)
    {
        propiedadesAUX = player.Value.CustomProperties;
        propiedadesAUX["jugadorSpawn"] = i;
        player.Value.SetCustomProperties(propiedadesAUX);
        i++;
    }
}
}
```