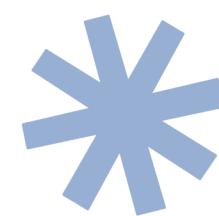


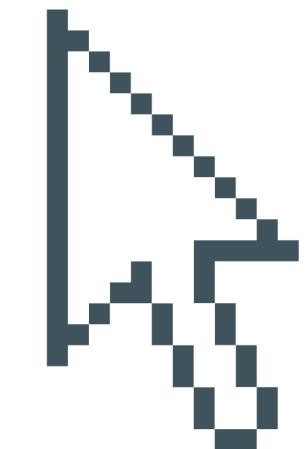


WOLVEDEN ACADEMY

NETWORKING AND MULTIPLAYER ONLINE GAMES



MULTIPLAYER NEW ERA



BY PONGSATHORN KIATTICHAOENPORN (MEW)

WEEK 7 : CONNECTING ONLINE 2

Connecting Online 2

Topic in this week

Workshop

- Relay Service Setup
- Allocating A Relay
- Joining A Relay
- Lobbies UI
- Creating Lobbies
- Joining Lobbies

Relay Service Setup

GAMING SERVICES Relay

Overview Use with Lobby

WorkshopMulti... GAMING SERVICES | MULTIPLAYER Relay Manage subscription Get started

Introduction to Relay | Unity Game... i :
1.00
0:00 / 2:57 • YouTube

Resources Documentation

Support Forum

WorkshopMulti... GAMING SERVICES Relay

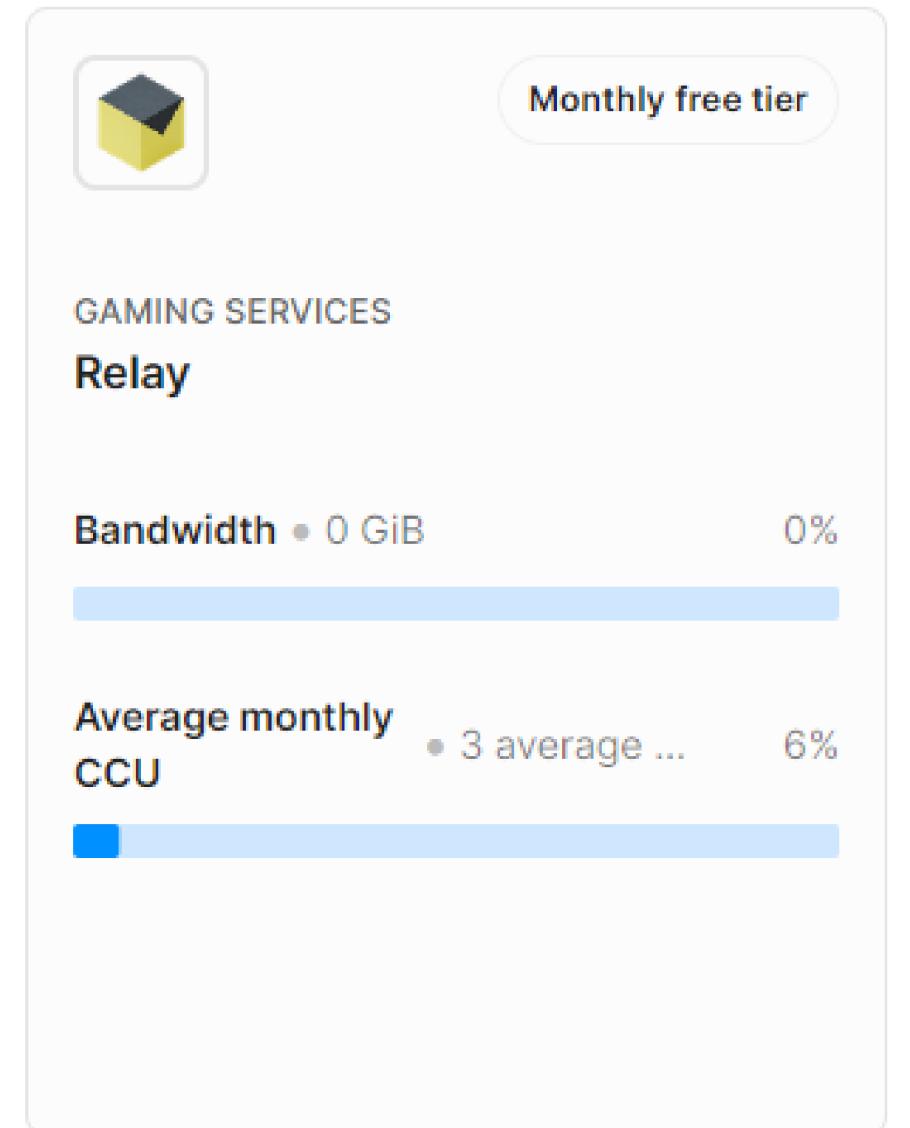
Setup guide

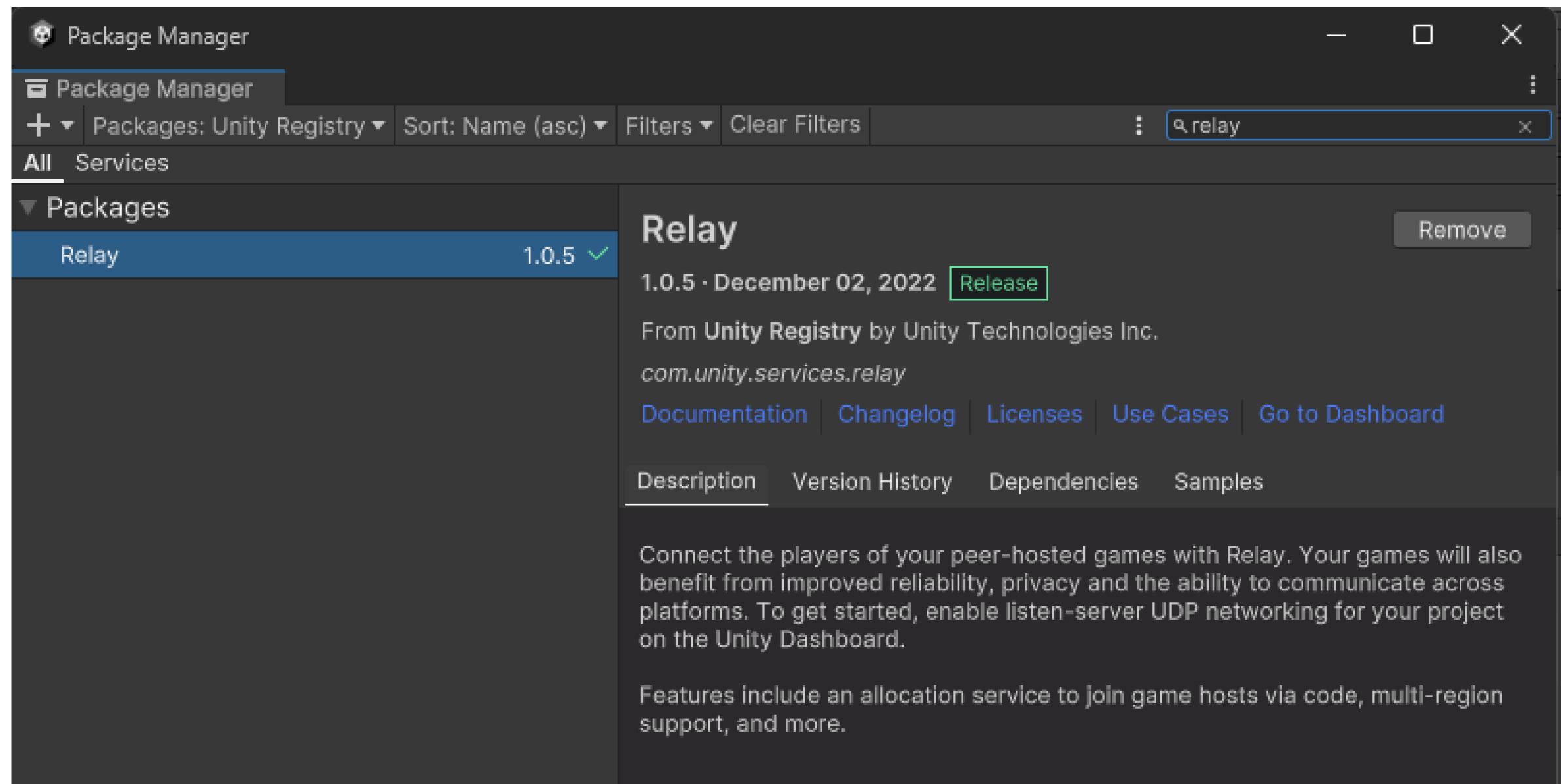
Reset guide

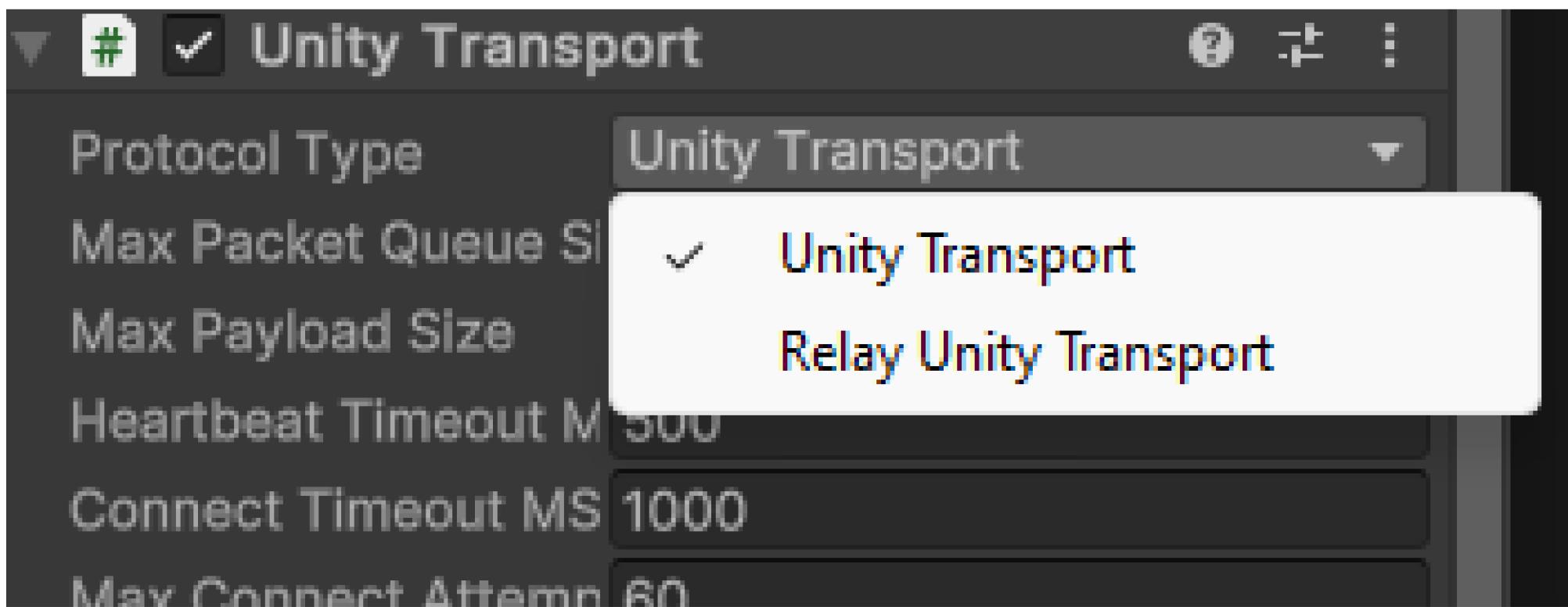
- Link Unity project
- Install package
- Import a sample

Relay has been successfully set up and automatically enabled.

Congratulations, you're ready to begin using Relay.

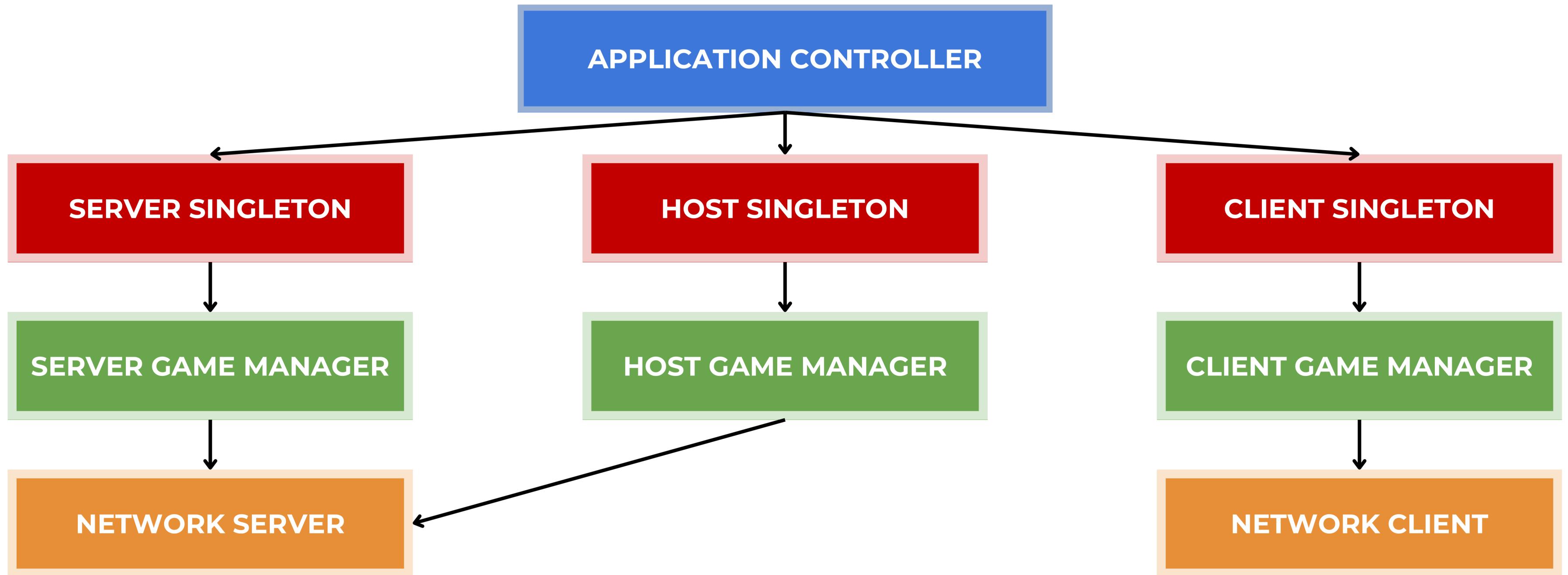






ในส่วนนี้เราไม่จำเป็นต้องปรับ เพราะเดียวเราจะเขียนโค้ดขึ้นมาเพื่อให้ได้กั้งสองแบบ

Application Structure



Allocating A Relay

```
public class ClientSingleton : MonoBehaviour
{
    private static ClientSingleton instance;
    3 references
    public ClientGameManager GameManager { get; private set; }

    0 references
    public static ClientSingleton Instance
    {
        get
        {
            if(instance != null) { return instance; }
            instance = FindFirstObjectByType<ClientSingleton>();

            if(instance == null)
            {
                Debug.LogError("No ClientSingleton in the scene!");
                return null;
            }
            return instance;
        }
    }
}
```

แก้โค้ดใน ClientSingleton

```
public class HostSingleton : MonoBehaviour
{
    private static HostSingleton instance;
    2 references
    public HostGameManager GameManager { get; private set; }

    1 reference
    public static HostSingleton Instance
    {
        get
        {
            if(instance != null) { return instance; }
            instance = FindFirstObjectByType<HostSingleton>();

            if(instance == null)
            {
                Debug.LogError("No HostSingleton in the scene!");
                return null;
            }
            return instance;
        }
    }
}
```

แก้โค้ดใน HostSingleton

```
① Unity Script (1 asset reference) | 5 references
6  [-] public class HostSingleton : MonoBehaviour
7  {
8      private static HostSingleton instance;
9
10     [SerializeField]
11     private HostGameManager gameManager;
```



```
[-] public class HostSingleton : MonoBehaviour
{
    private static HostSingleton instance;

    0 references
    public HostGameManager GameManager { get; private set; }

    public void CreateHost()
    {
        GameManager = new HostGameManager();
    }
}
```

```
[-] using System;
    using System.Collections;
    using System.Collections.Generic;
    using System.Threading.Tasks;
    using Unity.Services.Relay;
    using Unity.Services.Relay.Models;
    using UnityEngine;

    ④ Unity Script | 2 references
[-] public class HostGameManager : MonoBehaviour
{
    [-] {
        [-] private Allocation allocation;
        [-] private const int MaxConnections = 20;
        0 references
        [-] public async Task StartHostAsync()
        {
            [-] try
            {
                [-]     allocation = await Relay.Instance.CreateAllocationAsync(MaxConnections);
            }
            [-]         catch(Exception e)
            {
                [-]             Debug.Log(e);
                [-]             return;
            }
        }
    }
}
```

```
Unity Script | References
9 [-] public class HostGameManager : MonoBehaviour
10 {
11     private Allocation allocation;
12     private string joinCode;
13
14     private const int MaxConnections = 20;
15     0 references
16     public async Task StartHostAsync()
17     {
18         try
19         {
20             allocation = await Relay.Instance.CreateAllocationAsync(MaxConnections);
21         }
22         catch(Exception e)
23         {
24             Debug.Log(e);
25             return;
26         }
27         try
28         {
29             joinCode = await Relay.Instance.GetJoinCodeAsync(allocation.AllocationId);
30             Debug.Log(joinCode);
31         }
32         catch (Exception e)
33         {
34             Debug.Log(e);
35             return;
36         }
37     }
38 }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Threading.Tasks;
5  using Unity.Netcode;
6  using Unity.Netcode.Transports.UTP;
7  using Unity.Services.Relay;
8  using Unity.Services.Relay.Models;
9  using UnityEngine;
10
11     ② Unity Script | 2 references
12  [-]public class HostGameManager : MonoBehaviour
13  {
14      private Allocation allocation;
15      private string joinCode;
16
17      private const int MaxConnections = 20;
18
19      [-]public async Task StartHostAsync()
20      {
21          try ...
22
23          try
24          {
25              joinCode = await Relay.Instance.GetJoinCodeAsync(allocation.AllocationId);
26              Debug.Log(joinCode);
27          }
28          catch (Exception e)
29          {
30              Debug.Log(e);
31              return;
32          }
33
34      }
35
36  }
37
38
39  UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
40
41
42 }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Threading.Tasks;
5  using Unity.Netcode;
6  using Unity.Netcode.Transports.UTP;
7  using Unity.Networking.Transport.Relay;
8  using Unity.Services.Relay;
9  using Unity.Services.Relay.Models;
10 using UnityEngine;
11
12 ① Unity Script | 2 references
13  [-]public class HostGameManager : MonoBehaviour
14  {
15      private Allocation allocation;
16      private string joinCode;
17
18      [-]private const int MaxConnections = 20;
19
20      [-]public async Task StartHostAsync()
21      {
22          try{...}
23
24          try{...}
25
26          UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
27
28          RelayServerData relayServerData = new RelayServerData(allocation, "udp");
29          transport.SetRelayServerData(relayServerData);
30
31          NetworkManager.Singleton.StartHost();
32
33      }
34  }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Threading.Tasks;
5  using Unity.Netcode;
6  using Unity.Netcode.Transports.UTP;
7  using Unity.Networking.Transport.Relay;
8  using Unity.Services.Relay;
9  using Unity.Services.Relay.Models;
10 using UnityEngine;
11 using UnityEngine.SceneManagement;
12
12     ⚡ Unity Script | 2 references
13 public class HostGameManager : MonoBehaviour
14 {
15     private Allocation allocation;
16     private string joinCode;
17
18     private const int MaxConnections = 20;
19     private const string GameSceneName = "Game";
19     0 references
20     public async Task StartHostAsync()
21     {
22         try...
23
24         try...
25
26             UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
27
28             RelayServerData relayServerData = new RelayServerData(allocation, "udp");
29             transport.SetRelayServerData(relayServerData);
30
31             NetworkManager.Singleton.StartHost();
32
33             NetworkManager.Singleton.SceneManager.LoadScene(GameSceneName, LoadSceneMode.Single);
34
35     }
36 }
```

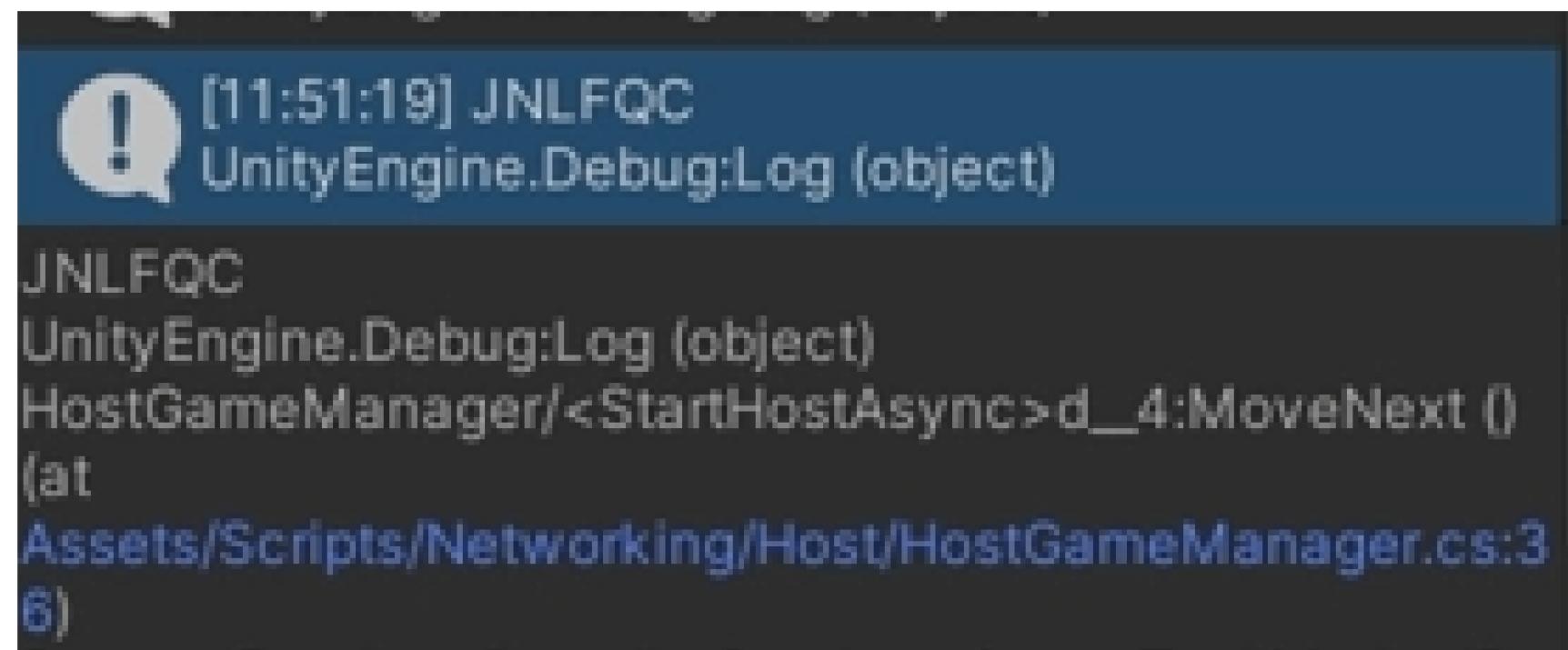
```
[-] using System.Collections;
    [-] using System.Collections.Generic;
    [-] using UnityEngine;

    ⚡ Unity Script (1 asset reference) | 0 references

[-] public class MainMenu : MonoBehaviour
    {
        [Lightbulb]
        ▶
        [
        ]
    }
```

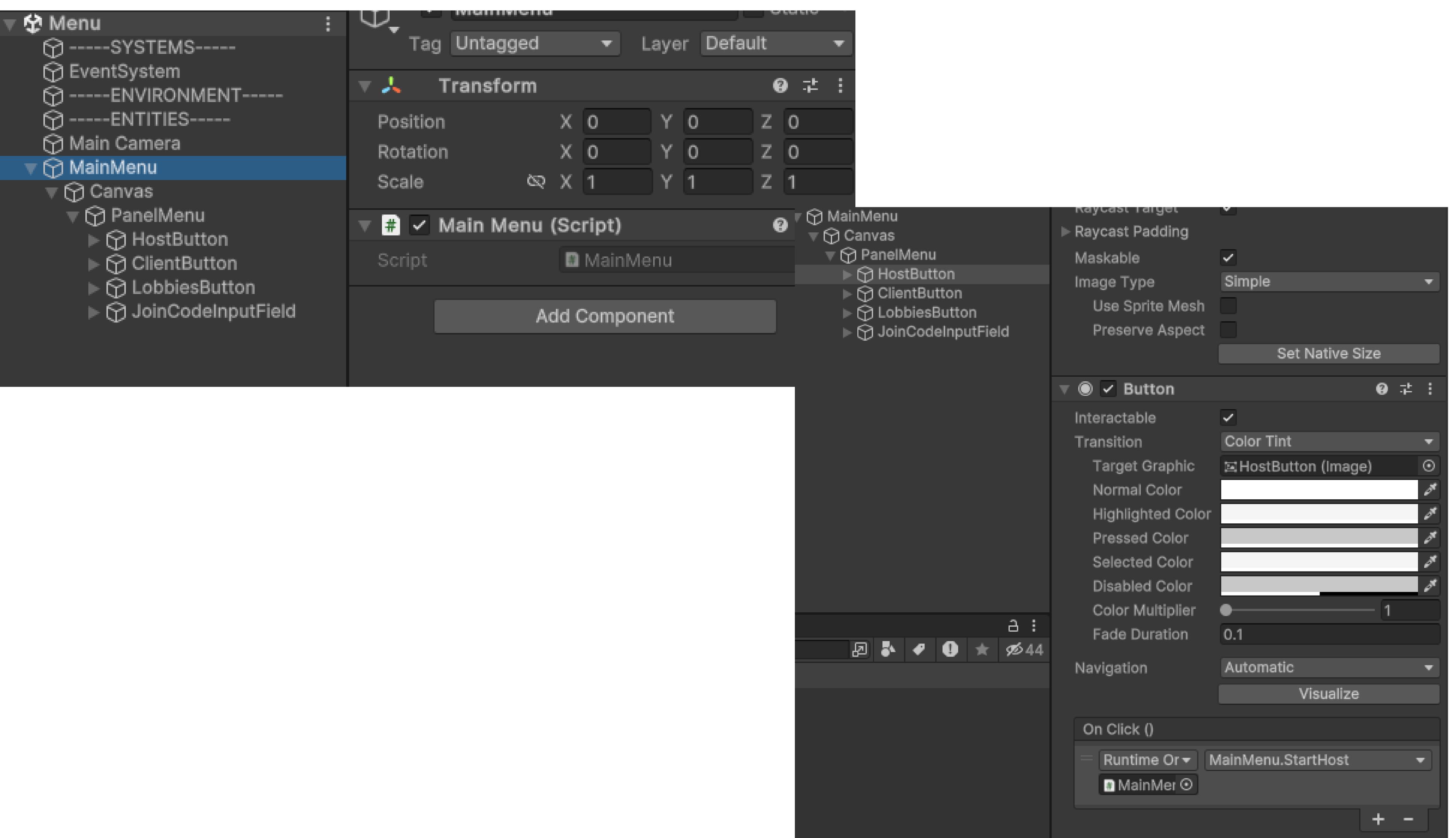
Hook Up To Host Button

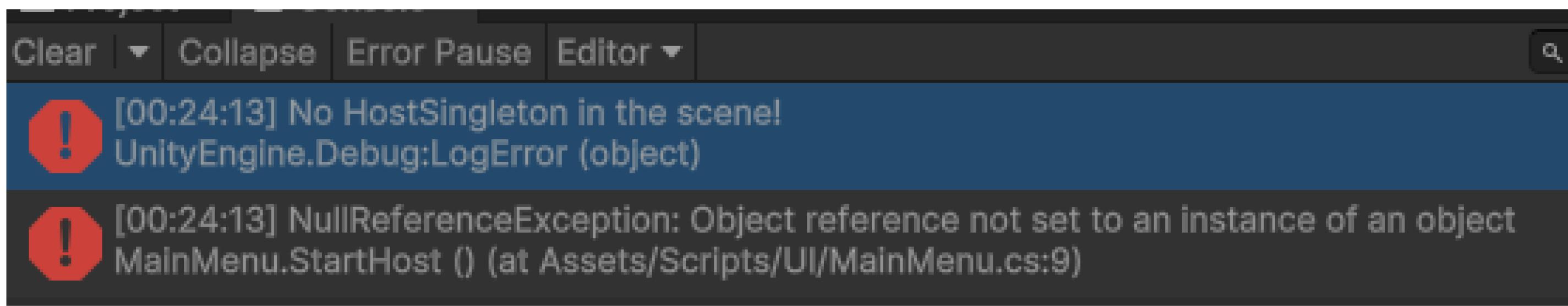
- Create a **StartHost** method in the **MainMenu** script
- Make this an **async void** since we'll need to await the **StartHostAsync** method
- Call **StartHostAsync**
- Hook up the button callback to this method



```
[-] using System.Collections;
[-]     using System.Collections.Generic;
[-]     using UnityEngine;

@ Unity Script (1 asset reference) | 0 references
[-] public class MainMenu : MonoBehaviour
{
    0 references
    [-]     public async void StartHost()
    {
        await HostSingleton.Instance.GameManager.StartHostAsync();
    }
}
```





```
public class ApplicationController : MonoBehaviour
{
    [SerializeField] private ClientSingleton clientPrefab;
    [SerializeField] private HostSingleton hostPrefab;

    // Start is called before the first frame update
    // Unity Message | 0 references
    private async void Start()...

    1 reference
    private async Task LaunchInMode(bool isDedicatedServer)
    {
        if (isDedicatedServer)...
        else
        {
            ClientSingleton clientSingleton = Instantiate(clientPrefab);
            bool authenticated = await clientSingleton.CreateClient();

            HostSingleton hostSingleton = Instantiate(hostPrefab);
            hostSingleton.CreateHost();
        }

        if (authenticated)
        {
            clientSingleton.GameManager.GoToMenu();
        }
    }
}
```

```
public class ApplicationController : MonoBehaviour
{
    [SerializeField] private ClientSingleton clientPrefab;
    [SerializeField] private HostSingleton hostPrefab;

    // Start is called before the first frame update
    // Unity Message | 0 references
    private async void Start()...

    1 reference
    private async Task LaunchInMode(bool isDedicatedServer)
    {
        if (isDedicatedServer)...
        else
        {
            HostSingleton hostSingleton = Instantiate(hostPrefab);
            hostSingleton.CreateHost();
        }

        ClientSingleton clientSingleton = Instantiate(clientPrefab);
        bool authenticated = await clientSingleton.CreateClient();

        if (authenticated)
        {
            clientSingleton.GameManager.GoToMenu();
        }
    }
}
```



Project Console

Clear ▾ Collapse Error Pause Editor ▾

[00:24:45] QosJob: send to 34.64.241.210:7778 took 0ms
UnityEngine.Debug:Log (object)

[00:24:45] QosJob: received 75 responses of 75/75 in 344ms waiting avg 5ms per response
UnityEngine.Debug:Log (object)

[00:24:45] QosJob: took 363ms to process 15 servers
UnityEngine.Debug:Log (object)

[00:24:45] best region is asia-southeast1
UnityEngine.Debug:Log (object)

[00:24:46] CFKHQC
UnityEngine.Debug:Log (object)

Joining A Relay

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;

④ Unity Script (1 asset reference) | 0 references
public class MainMenu : MonoBehaviour
{
    [SerializeField] private TMP_InputField joinCodeField;
    0 references
    public async void StartHost()
    {
        await HostSingleton.Instance.GameManager.StartHostAsync();
    }

    0 references
    public async void StartClient()
    {
        await ClientSingleton.Instance.GameManager.StartClientAsync(joinCodeField.text)
    }
}
```

[Controller.cs](#)[InputReader.cs](#)[HostGameManager.cs](#)[ClientSingleton.cs](#)[HostSingleton.cs](#)[ClientGameManager.cs](#)

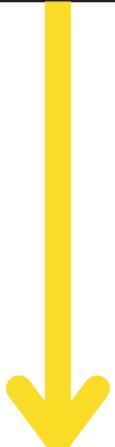
HostGameManager

StartHostAsync()

try...

```
UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
```

```
RelayServerData relayServerData = new RelayServerData(allocation, "udp");  
transport.SetRelayServerData(relayServerData);
```



```
UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
```

```
RelayServerData relayServerData = new RelayServerData(allocation, "dtls");  
transport.SetRelayServerData(relayServerData);
```

The screenshot shows a code editor with multiple tabs at the top: 'menu.cs', 'ApplicationController.cs', 'ClientGameManager.cs*', 'InputReader.cs', and 'HostGameManager.cs'. The 'ClientGameManager.cs*' tab is active, highlighted with a purple bar. The code editor displays the following C# code:

```
28
29     public async Task StartClientAsync(string joinCode)
30     {
31         // ...
32     }
33 }
34
```

A green vertical line highlights the opening brace of the method definition at line 31. A small orange lightbulb icon is positioned next to the line 31 brace, indicating a potential issue or suggestion. The status bar at the bottom of the editor window shows the text 'ClientGameManager'.

Join The Allocation

- Create a **private JoinAllocation allocation;** field

- Make the call to

Relay.Instance.JoinAllocationAsync

asynchronously

- Wrap this call in a try-catch block like we have done previously

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Threading.Tasks;
5  using Unity.Services.Core;
6  using Unity.Services.Relay;
7  using Unity.Services.Relay.Models;
8  using UnityEngine;
9  using UnityEngine.SceneManagement;
10
11 2 references
12 [-]public class ClientGameManager
13 {
14     private JoinAllocation allocation;
15     private const string MenuSceneName = "Menu";
16     1 reference
17     public async Task<bool> InitAsync()...
18
19     1 reference
20     public void GoToMenu()...
21
22     1 reference
23     public async Task StartClientAsync(string joinCode)
24     {
25         try
26         {
27             allocation = await Relay.Instance.JoinAllocationAsync(joinCode);
28         }
29         catch (Exception e)
30         {
31             Debug.Log(e);
32             return;
33         }
34     }
35 }
```

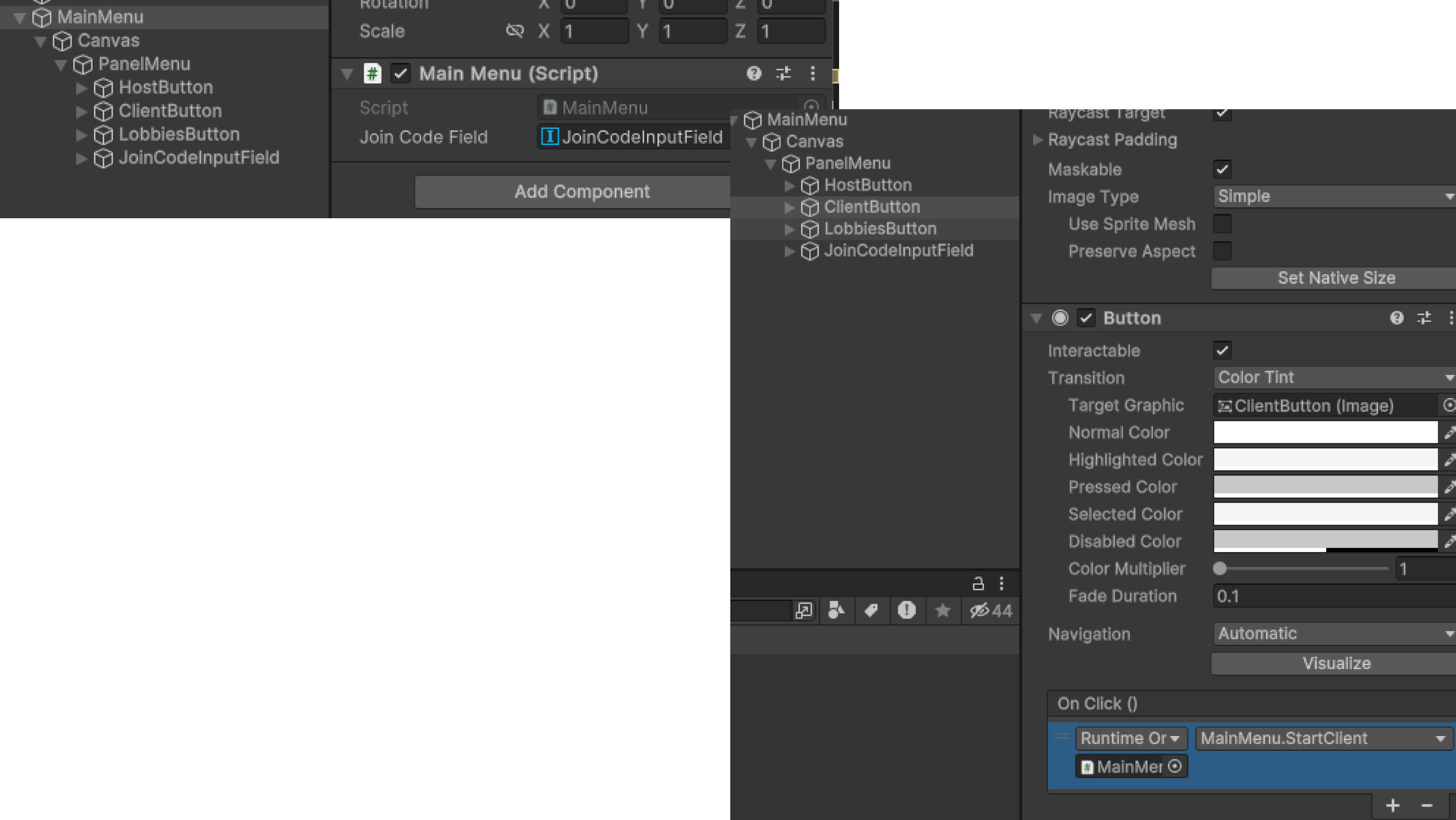
ApplicationController.cs ClientGameManager.cs* InputReader.cs HostGameManager.cs ✘ ClientSingleton.cs HostSingleton.cs

Sharp

+ HostGameManager

- StartHostAsync()

3
9
1 reference
public const int MaxConnections = 20;
private const string GameSceneName = "Game";
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
55310
55311
55312
55313
55314
55315
55316
55317
55318
55319
55320
55321
55322
55323
55324
55325
55326
55327
55328
55329
55330
55331
55332
55333
55334
55335
55336
55337
55338
55339
55340
55341
55342
55343
55344
55345
55346
55347
55348
55349
55350
55351
55352
55353
55354
55355
55356
55357
55358
55359
55360
55361
55362
55363
55364
55365
55366
55367
55368
55369
55370
55371
55372
55373
55374
55375
55376
55377
55378
55379
55380
55381
55382
55383
55384
55385
55386
55387
55388
55389
55390
55391
55392
55393
55394
55395
55396
55397
55398
55399
553100
553101
553102
553103
553104
553105
553106
553107
553108
553109
553110
553111
553112
553113
553114
553115
553116
553117
553118
553119
553120
553121
553122
553123
553124
553125
553126
553127
553128
553129
553130
553131
553132
553133
553134
553135
553136
553137
553138
553139
553140
553141
553142
553143
553144
553145
553146
553147
553148
553149
553150
553151
553152
553153
553154
553155
553156
553157
553158
553159
553160
553161
553162
553163
553164
553165
553166
553167
553168
553169
553170
553171
553172
553173
553174
553175
553176
553177
553178
553179
553180
553181
553182
553183
553184
553185
553186
553187
553188
553189
553190
553191
553192
553193
553194
553195
553196
553197
553198
553199
553200
553201
553202
553203
553204
553205
553206
553207
553208
553209
553210
553211
553212
553213
553214
553215
553216
553217
553218
553219
553220
553221
553222
553223
553224
553225
553226
553227
553228
553229
553230
553231
553232
553233
553234
553235
553236
553237
553238
553239
553240
553241
553242
553243
553244
553245
553246
553247
553248
553249
553250
553251
553252
553253
553254
553255
553256
553257
553258
553259
553260
553261
553262
553263
553264
553265
553266
553267
553268
553269
553270
553271
553272
553273
553274
553275
553276
553277
553278
553279
553280
553281
553282
553283
553284
553285
553286
553287
553288
553289
553290
553291
553292
553293
553294
553295
553296
553297
553298
553299
553300
553301
553302
553303
553304
553305
553306
553307
553308
553309
553310
553311
553312
553313
553314
553315
553316
553317
553318
553319
553320
553321
553322
553323
553324
553325
553326
553327
553328
553329
553330
553331
553332
553333
553334
553335
553336
553337
553338
553339
5533310
5533311
5533312
5533313
5533314
5533315
5533316
5533317
5533318
5533319
55333110
55333111
55333112
55333113
55333114
55333115
55333116
55333117
55333118
55333119
553331110
553331111
553331112
553331113
553331114
553331115
553331116
553331117
553331118
553331119
5533311110
5533311111
5533311112
5533311113
5533311114
5533311115
5533311116
5533311117
5533311118
5533311119
55333111110
55333111111
55333111112
55333111113
55333111114
55333111115
55333111116
55333111117
55333111118
55333111119
553331111110
553331111111
553331111112
553331111113
553331111114
553331111115
553331111116
553331111117
553331111118
553331111119
5533311111110
5533311111111
5533311111112
5533311111113
5533311111114
5533311111115
5533311111116
5533311111117
5533311111118
5533311111119
55333111111110
55333111111111
55333111111112
55333111111113
55333111111114
55333111111115
55333111111116
55333111111117
55333111111118
55333111111119
553331111111110
553331111111111
553331111111112
553331111111113
553331111111114
553331111111115
553331111111116
553331111111117
553331111111118
553331111111119
5533311111111110
5533311111111111
5533311111111112
5533311111111113
5533311111111114
5533311111111115
5533311111111116
5533311111111117
5533311111111118
5533311111111119
55333111111111110
55333111111111111
55333111111111112
55333111111111113
55333111111111114
55333111111111115
55333111111111116
55333111111111117
55333111111111118
55333111111111119
553331111111111110
553331111111111111
553331111111111112
553331111111111113
553331111111111114
553331111111111115
553331111111111116
553331111111111117
553331111111111118
553331111111111119
5533311111111111110
5533311111111111111
5533311111111111112
5533311111111111113
5533311111111111114
5533311111111111115
5533311111111111116
5533311111111111117
5533311111111111118
5533311111111111119
55333111111111111110
55333111111111111111
55333111111111111112
55333111111111111113
55333111111111111114
55333111111111111115
55333111111111111116
55333111111111111117
55333111111111111118
55333111111111111119
553331111111111111110
553331111111111111111
553331111111111111112
553331



Build Settings

Build Settings

Scenes In Build

- Scenes/NetBootstrap
- Scenes/Menu
- Scenes/Game

Add Open Scenes

Platform

- Windows, Mac, Linux
- Android
- iOS
- WebGL
- Dedicated Server
- PS4
- PS5
- Universal Windows Platform

Windows, Mac, Linux

Target Platform: Windows

Architecture: Intel 64-bit

Build and Run on: Local Machine

Copy PDB files:

Create Visual Studio Solution:

Development Build:

Autoconnect Profiler:

Deep Profiling:

Script Debugging:

Compression Method: Default

Asset Import Overrides

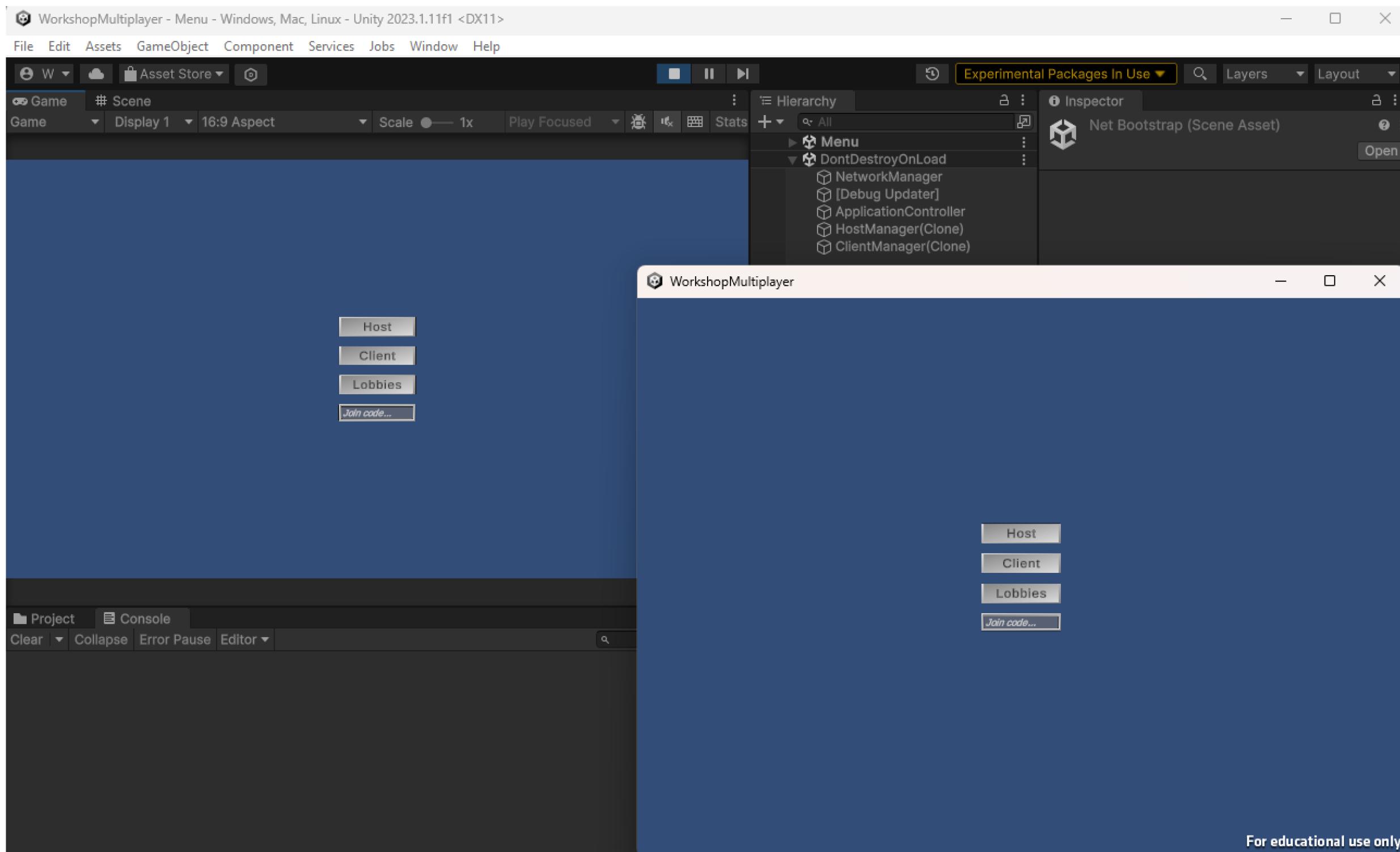
Max Texture Size: No Override

Texture Compression: No Override

Learn about Unity Build Automation

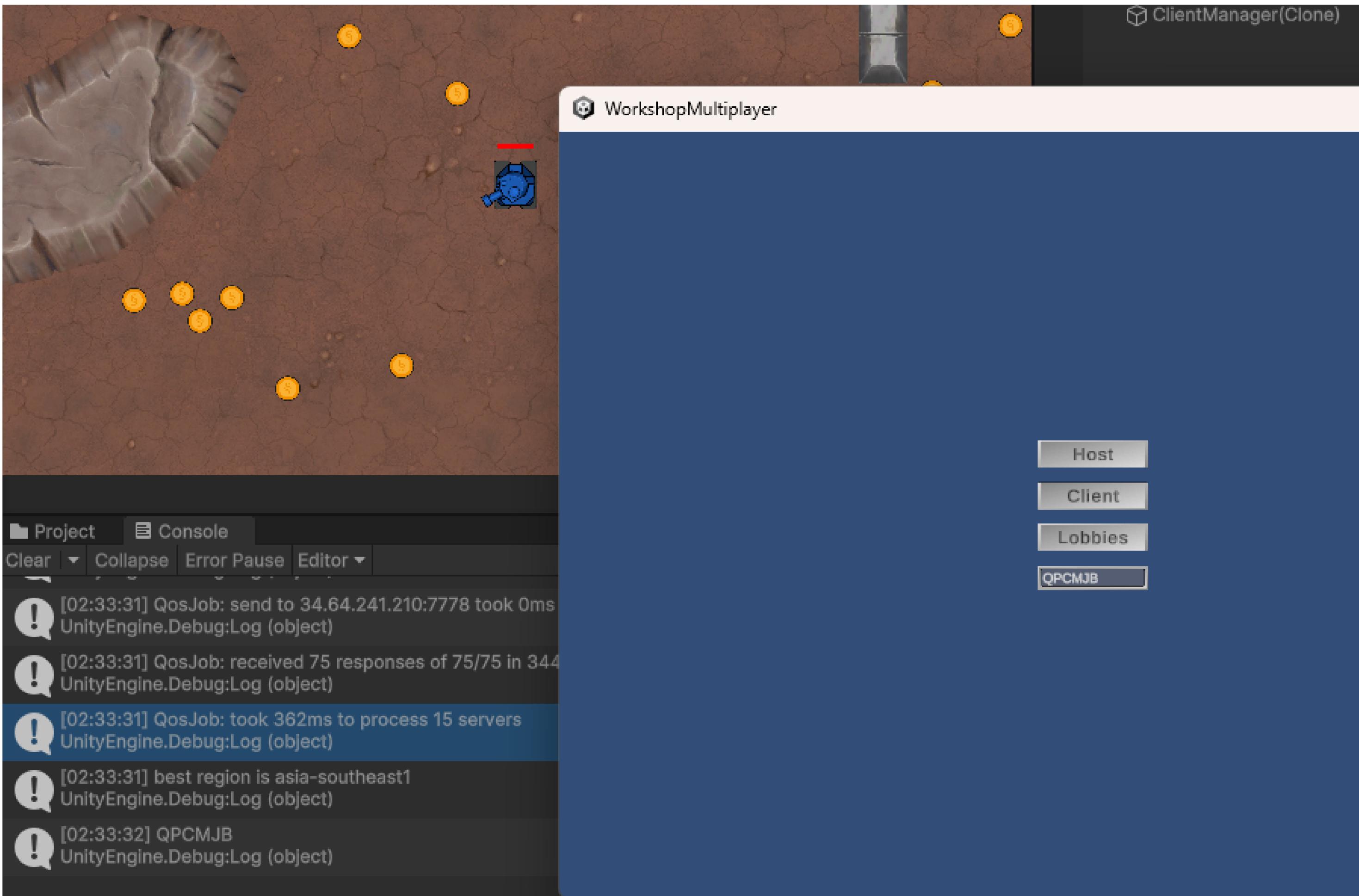
Player Settings... Build | Build And Run

Build เพื่อกำกับการทดสอบ



ให้ผู้ใช้งานเป็น Host
แล้วดูรูปที่ถูกสร้างขึ้นมา

ให้ผู้ใช้งานเป็น Client
ให้นำรูปที่ได้จาก Host มาใช้



Lobbies UI

 User Generated Content Cloud Diagnostics Advanced Cloud Content Delivery Unity Ads User Acquisition Unity Ads Monetization Game Server Hosting Safe Voice Vivox Voice and Text Chat Remote Config DevOps Push Notifications Cloud Code Relay Player Authentication

Overview

Use with Lobby

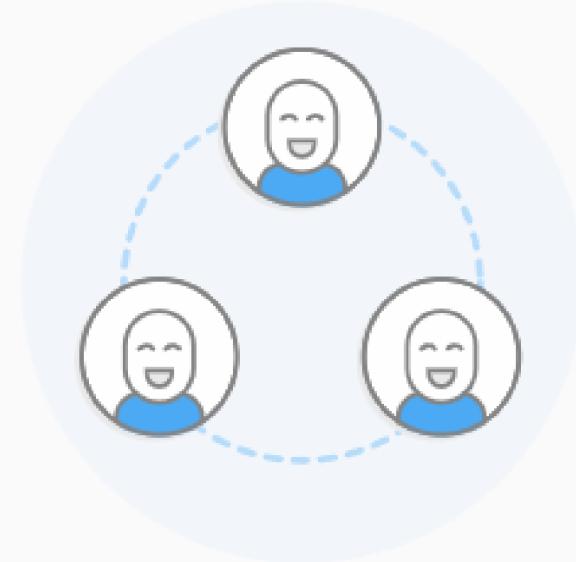
Using Relay with Lobby

Relay and Lobby work best together

Improve your player experience with Lobby and Relay together. Send service-to-service notifications when a player disconnects from the game and automatically remove players disconnected from Relay.

What you get with Lobby:

- **Public lobby management and discoverability**
Allow players to create lobbies that other players can find and join.
- **Out-of-the-box Quick Join functionality**
Allow players to easily join the best lobbies for their criteria.
- **Private lobby support**
Enable Private Match so players can create private lobbies and invite their friends.

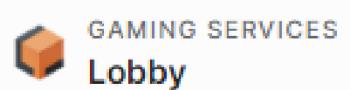
[Open Lobby](#)[Lobby documentation !\[\]\(73f697ce1ea97739c0bdb25596717315_img.jpg\)](#)

Documentation

Usage

Setup guide

About



GAMING SERVICES
Lobby

Overview

Config

Use with Relay

WorkshopMultiplayer (2) ▾

GAMING SERVICES | MULTIPLAYER

Lobby

Manage subscription

Launch

Introduction to Lobby | Unity Gaming Services

1.00

Share

Info

0:00 / 3:15 • Introduction

YouTube

Resources

Documentation

Support

Forum

[Overview](#) [Pricing](#) [Use With](#)

Lobby allows your players to connect before or during a game session. Players create public lobbies using simple game attributes which other players can then search, discover, and join. Invite-only lobbies also allow players to create private spaces for select participants only.

Start building for free

Our pricing model is designed to fit the needs of developers of any size. Lobby offers a free tier that covers your pre-development needs and pricing based on consumption – so you only pay as your game grows.

Documentation

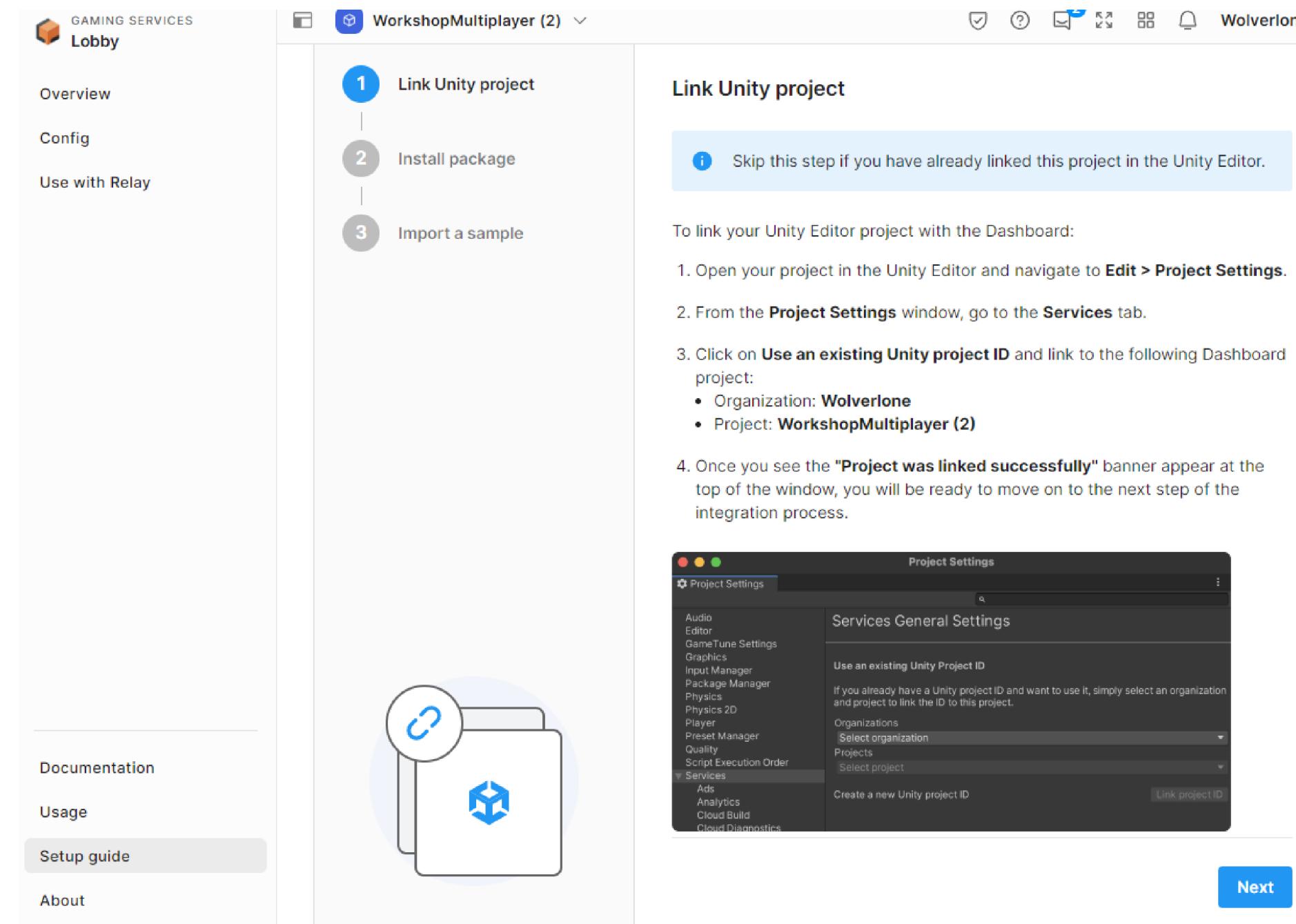
Usage

Setup guide

About

Set Up Lobby

- Enable the **Lobby** service
- Follow the instructions to set up the service in your project



All Services

▼ Packages

Lobby	1.1.2	✓
-------	-------	---

Lobby

1.1.2 · December 12, 2023 [Release](#)

From Unity Registry by Unity Technologies Inc.

com.unity.services.lobby

[Documentation](#) | [Changelog](#) | [Licenses](#) | [Use Cases](#) | [Go to Dashboard](#)

[Description](#) [Version History](#) [Dependencies](#) [Samples](#)

Enable players to find, create, and join lobbies with Lobby. Add Lobby to your multiplayer game to empower players to create the gaming experiences they want. Features include Quick Join, Public Lobby, and Private Match to allow flexibility in how players want to game with each other.

Remove

A screenshot of the Unity Registry interface. On the left, there's a sidebar with 'All Services' and a 'Packages' section expanded, showing a single entry for 'Lobby' at version 1.1.2. The main panel on the right displays detailed information about the 'Lobby' package. It includes the package name 'Lobby', its version '1.1.2', a release date of 'December 12, 2023', and a link to the 'Release' note. It also shows the source as 'From Unity Registry by Unity Technologies Inc.' and the namespace as 'com.unity.services.lobby'. Below this, there are links to 'Documentation', 'Changelog', 'Licenses', 'Use Cases', and 'Go to Dashboard'. At the bottom, there are tabs for 'Description', 'Version History', 'Dependencies', and 'Samples'. A large paragraph at the bottom describes the package's purpose: enabling players to find, create, and join lobbies with features like Quick Join, Public Lobby, and Private Match.

GAMING SERVICES

Lobby

WorkshopMultiplayer (2)

?

Wolverlone W

Overview

Config

Use with Relay

Setup guide

Reset guide

- Link Unity project
- Install package
- Import a sample

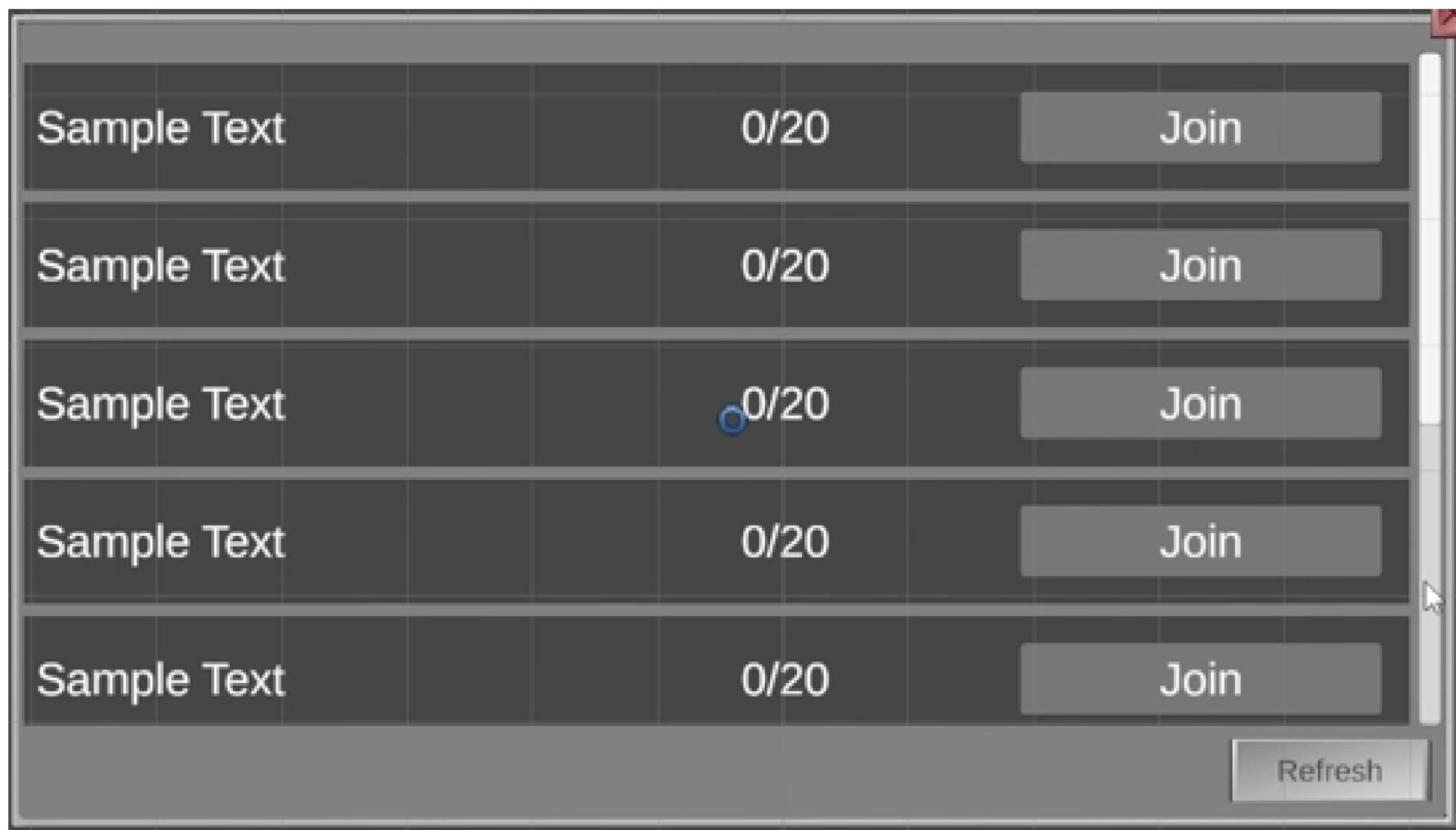
Setup complete

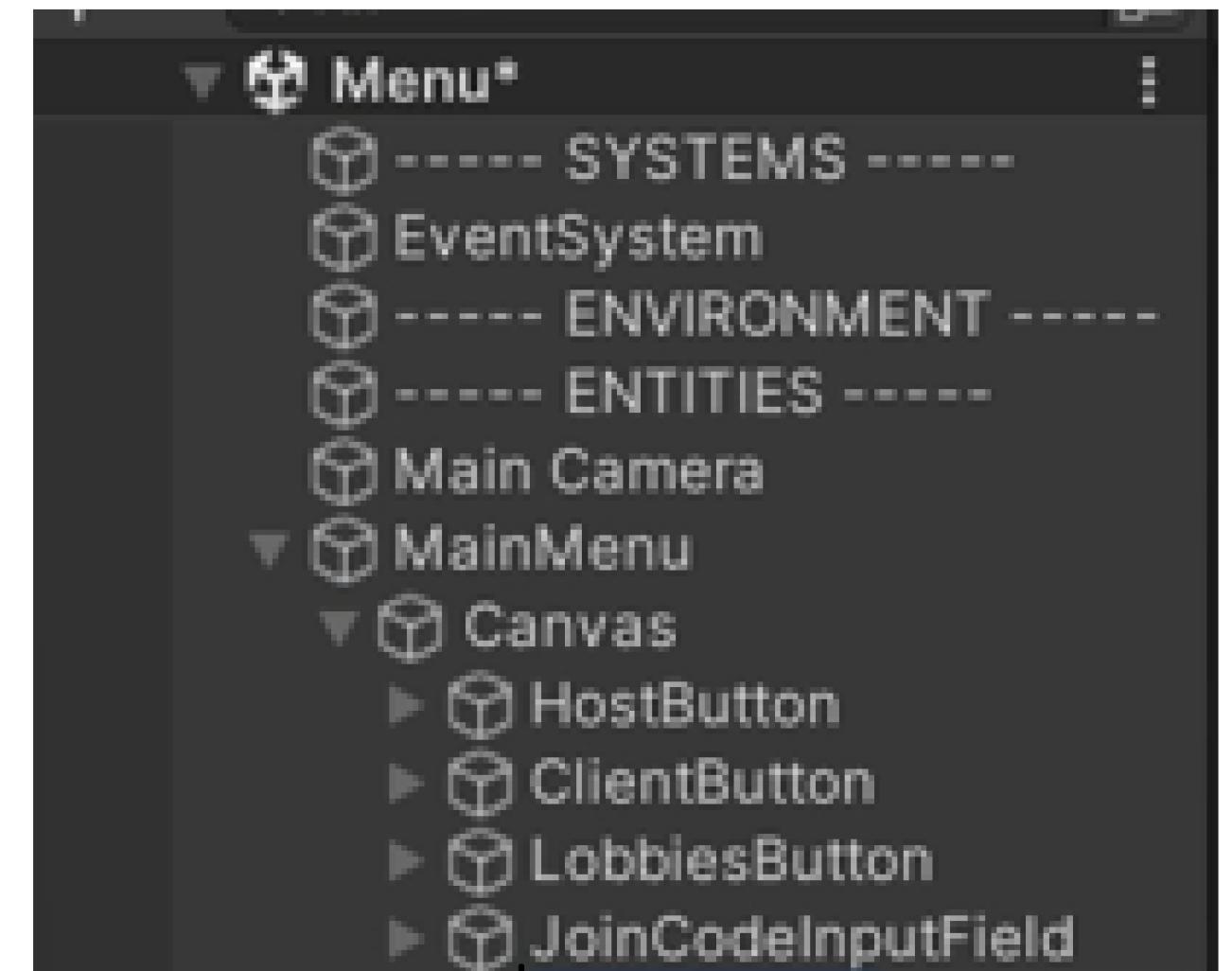
i Lobby has been successfully set up and automatically enabled.

Congratulations, you're ready to begin using Lobby.

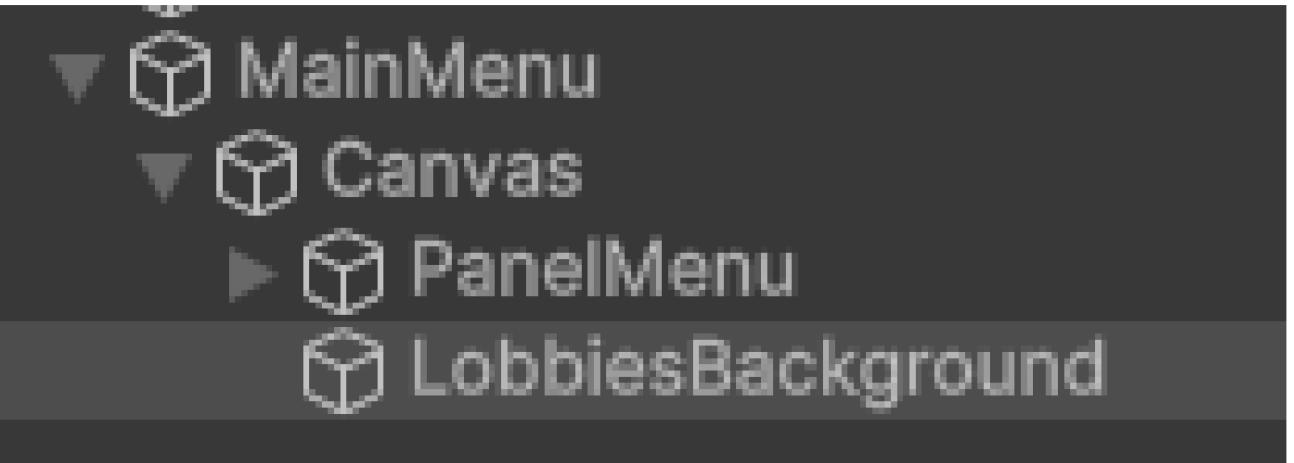
What's next?

ເຕີຍມສ້າງຫັໍາ Lobby





ไปที่จาก Menu



The Inspector panel shows the properties of the selected **LobbiesBackground** object. The object is a **Rect Transform** component attached to a **Image** component.

Rect Transform Properties:

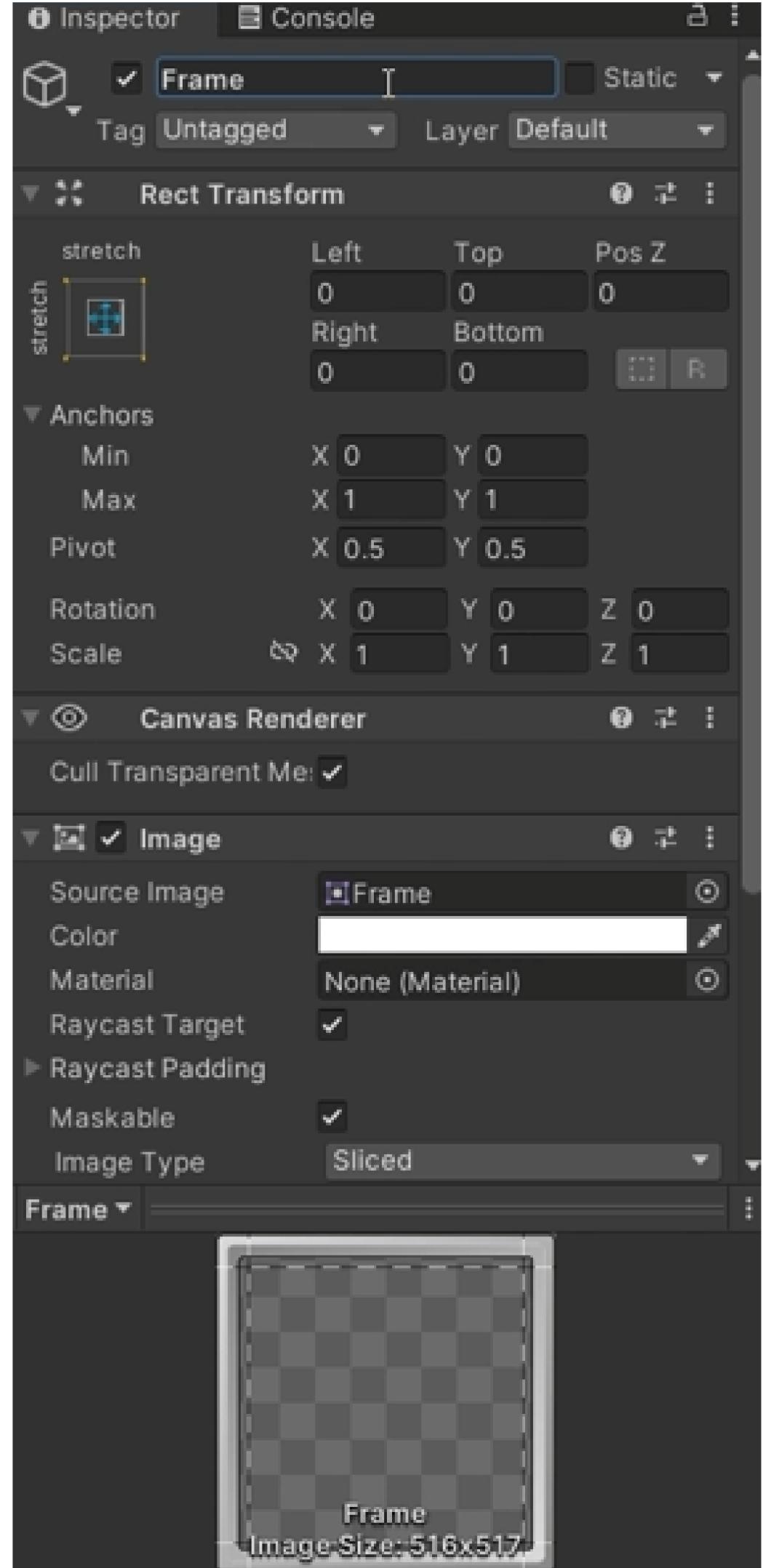
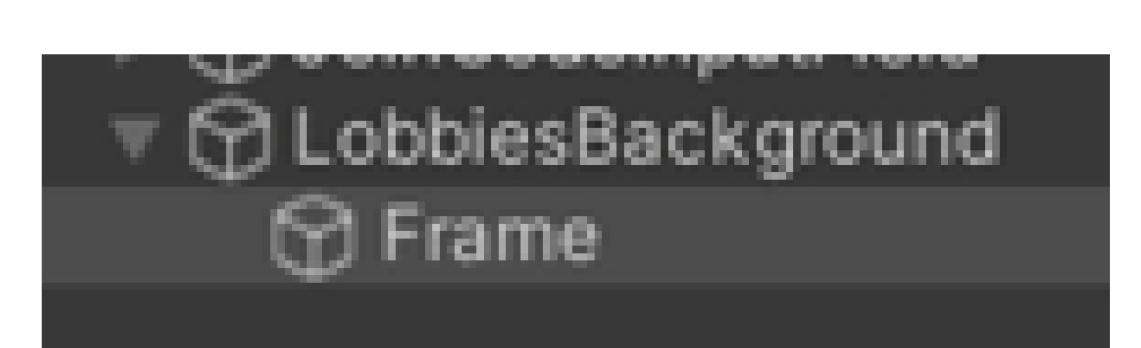
- Scale:** 0.49x
- Play Focused:** Enabled
- Custom Anchors:** Left: 0, Top: 0, Pos Z: 0; Right: 0, Bottom: 0
- Anchors:**
 - Min: X: 0.2, Y: 0.2
 - Max: X: 0.8, Y: 0.8
 - Pivot: X: 0.5, Y: 0.5
- Rotation:** X: 0, Y: 0, Z: 0
- Scale:** X: 1, Y: 1, Z: 1

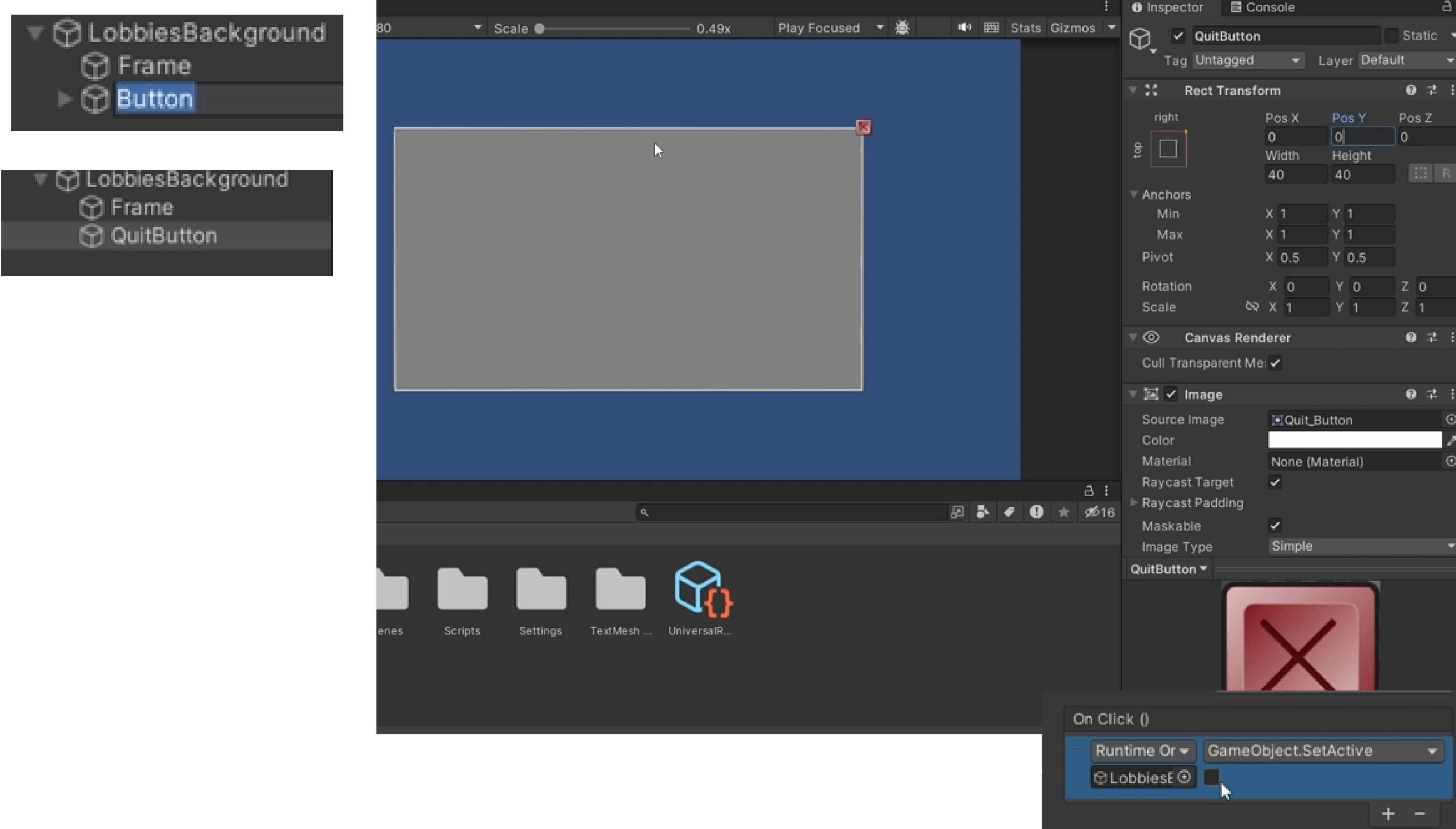
Canvas Renderer Properties:

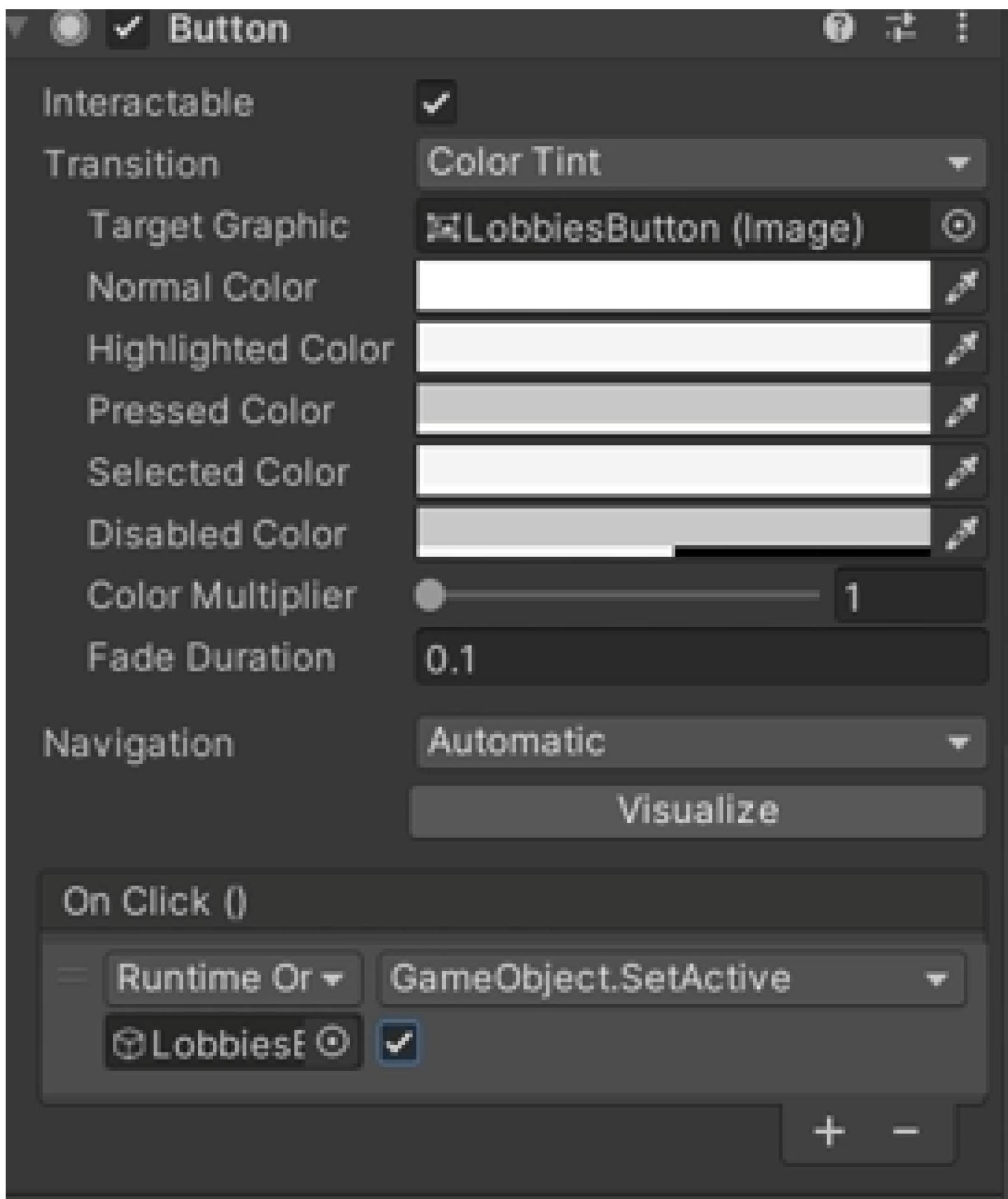
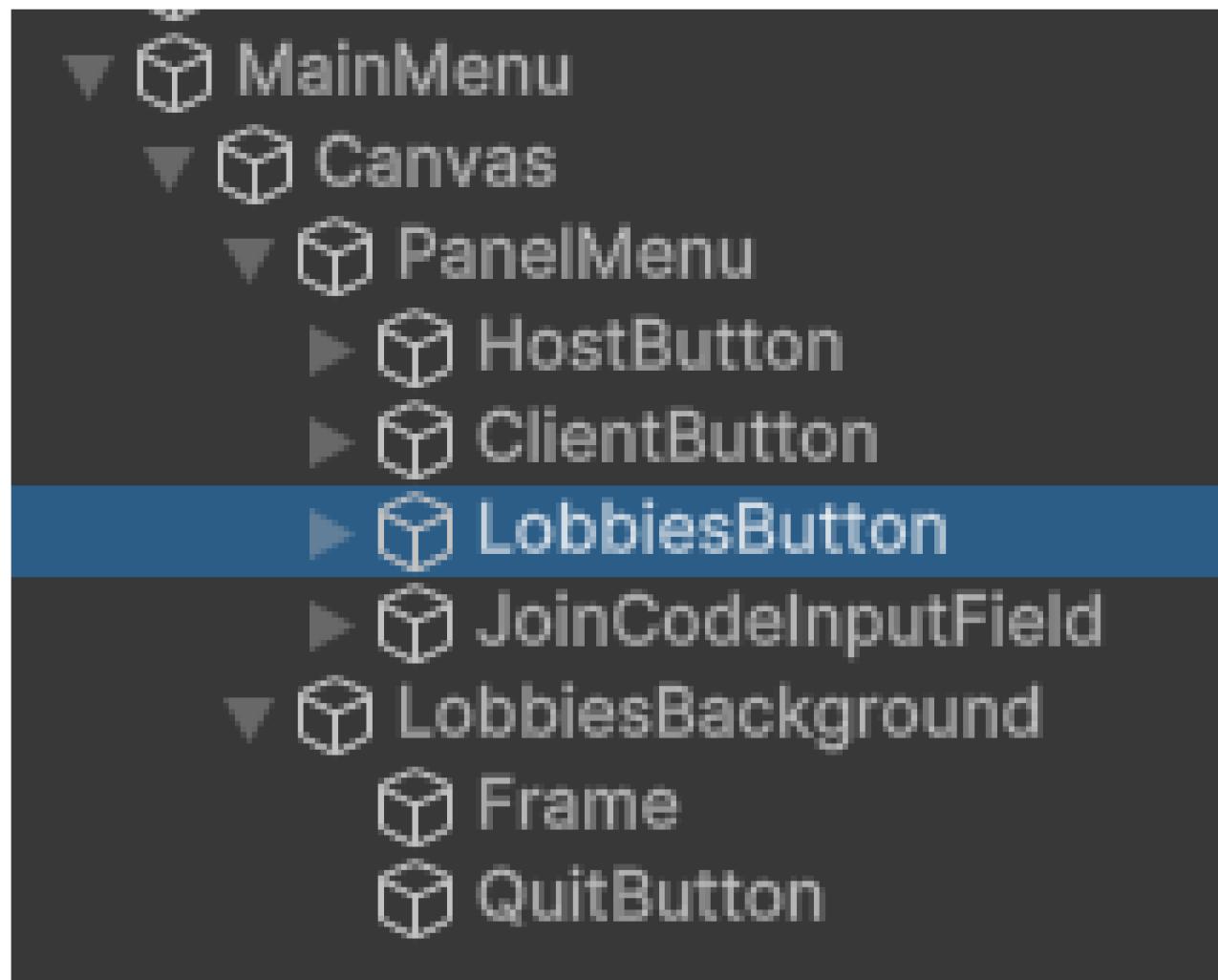
- Cull Transparent Mesh:** Enabled

Image Properties:

- Source Image:** None (Sprite)
- Color:** Gray
- Material:** None (Material)
- Raycast Target:** Enabled
- Raycast Padding:** (List item)







The screenshot shows the Unity Editor's Inspector and Component panels for a UI element named "RefreshButton".

Inspector Panel (Left):

- Source Image: Plate
- Color: None (Material)
- Raycast Target: checked
- Raycast Padding: checked
- Maskable: checked
- Image Type: Simple

Component Panel (Middle):

- Transform:** Static, Tag: Untagged, Layer: Default
- Rect Transform:**
 - Position: Pos X: -20, Pos Y: 20, Pos Z: 0
 - Size: Width: 160, Height: 50
 - Anchor: right, bottom
 - Min: X: 1, Y: 0
 - Max: X: 1, Y: 0
 - Pivot: X: 1, Y: 0
 - Rotation: X: 0, Y: 0, Z: 0
 - Scale: Local X: 1, Y: 1, Z: 1

Text Component Panel (Right):

- Text Style: Normal
- Main Settings:**
 - Font Asset: LiberationSans SDF (TMP)
 - Material Preset: LiberationSans SDF Material
 - Font Style: B I U S ab AB SC
 - Font Size: 24
 - Auto Size: checked
 - Vertex Color: [Color Swatch]
 - Color Gradient: [Color Swatch]
 - Override Tags: [checkbox]
 - Spacing Options (emCharacter: 0, Word: 0, Line: 0, Paragraph: 0)
- Alignment: [Horizontal Alignment: Left, Vertical Alignment: Top]

The screenshot shows the Unity Editor's Hierarchy and Network panels.

Hierarchy Panel (Left):

- LobbiesBackground
 - Frame
 - QuitButton
 - RefreshButton
 - RefreshText

Network Panel (Bottom Right):

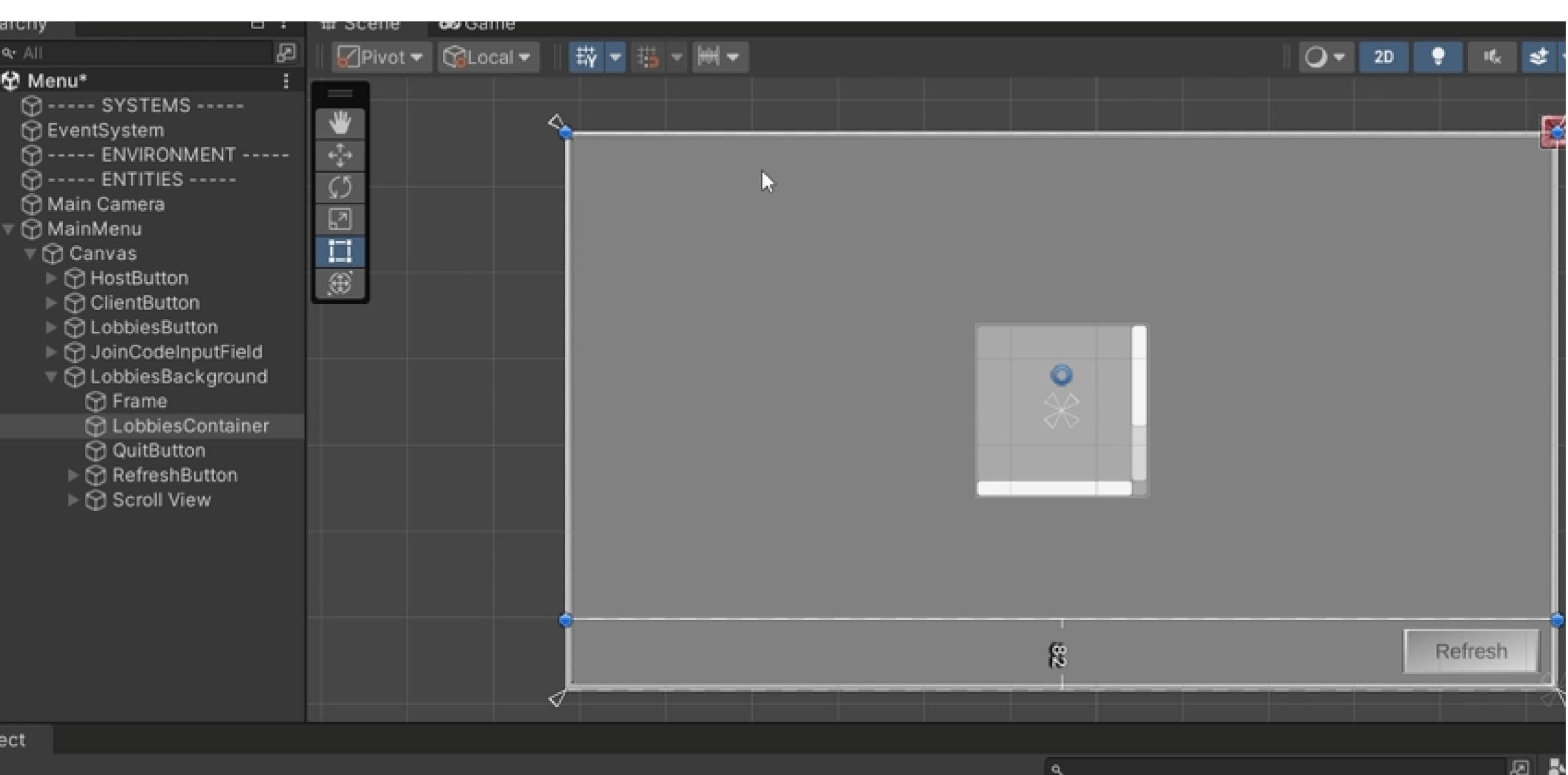
- A network connection diagram for the "Refresh" button. It shows a sequence of nodes: a white square, a blue circle, a yellow rectangle, another white square, a blue circle, and a final white square. The word "Refresh" is written below the yellow rectangle node.

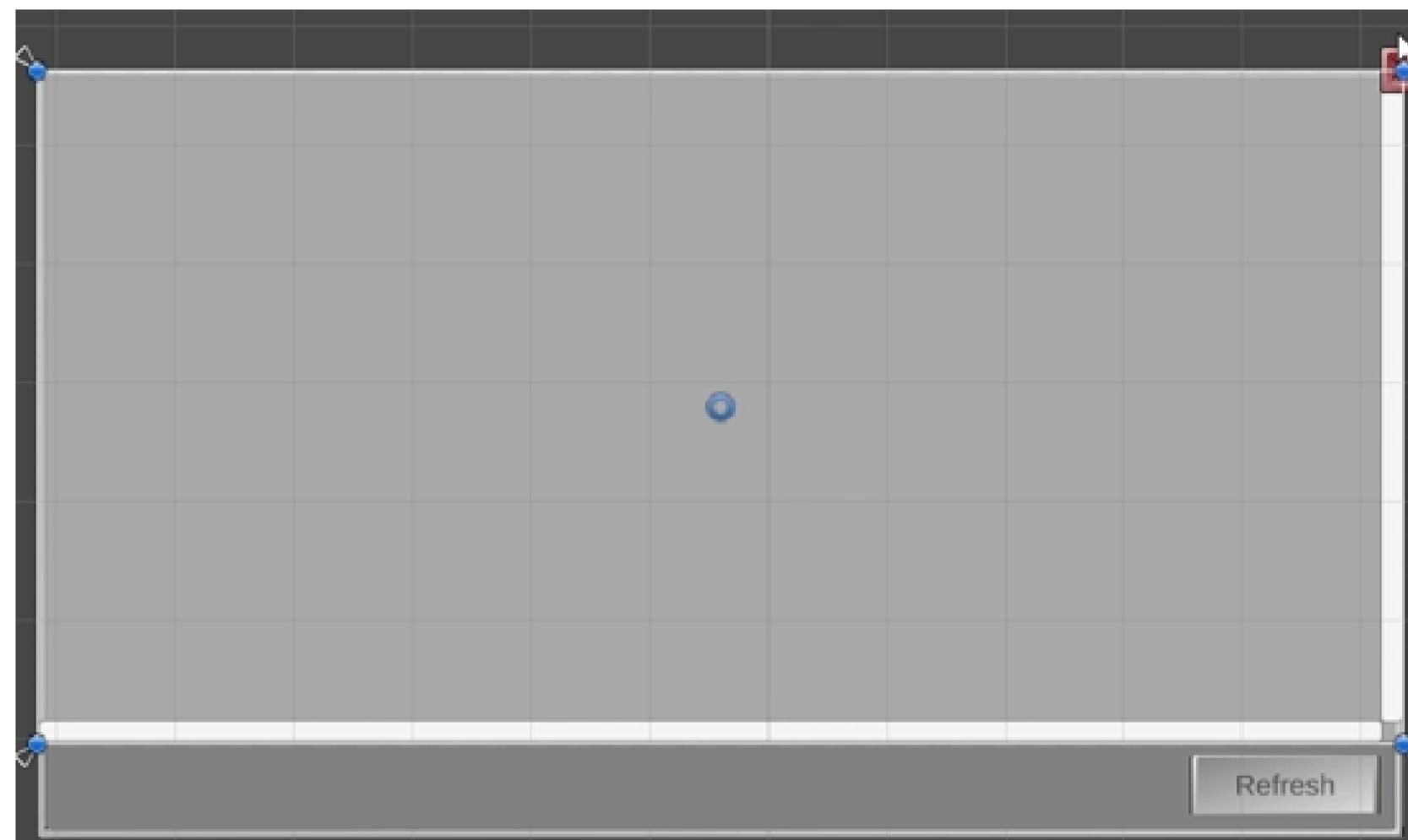
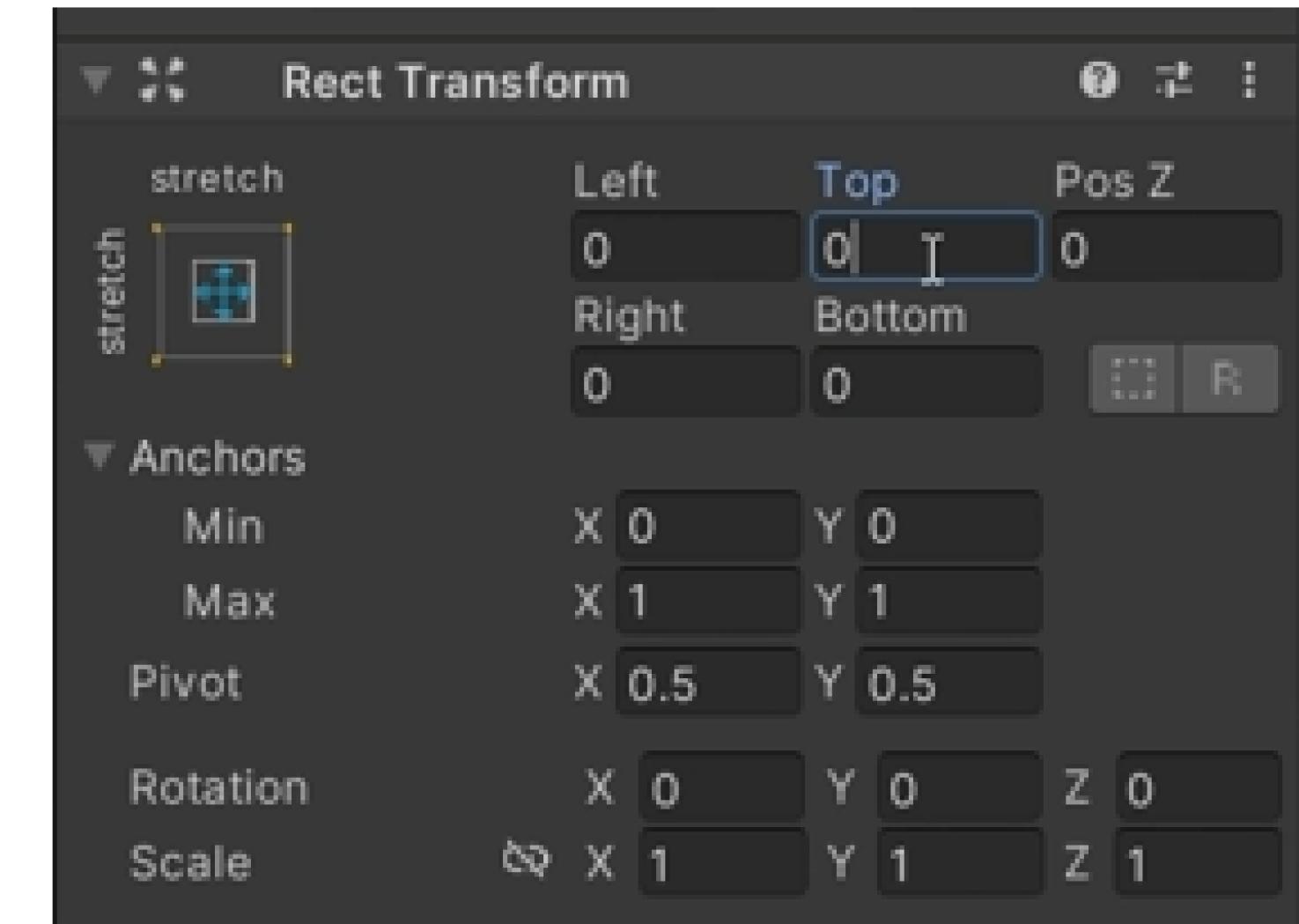
▼ LobbiesBackground

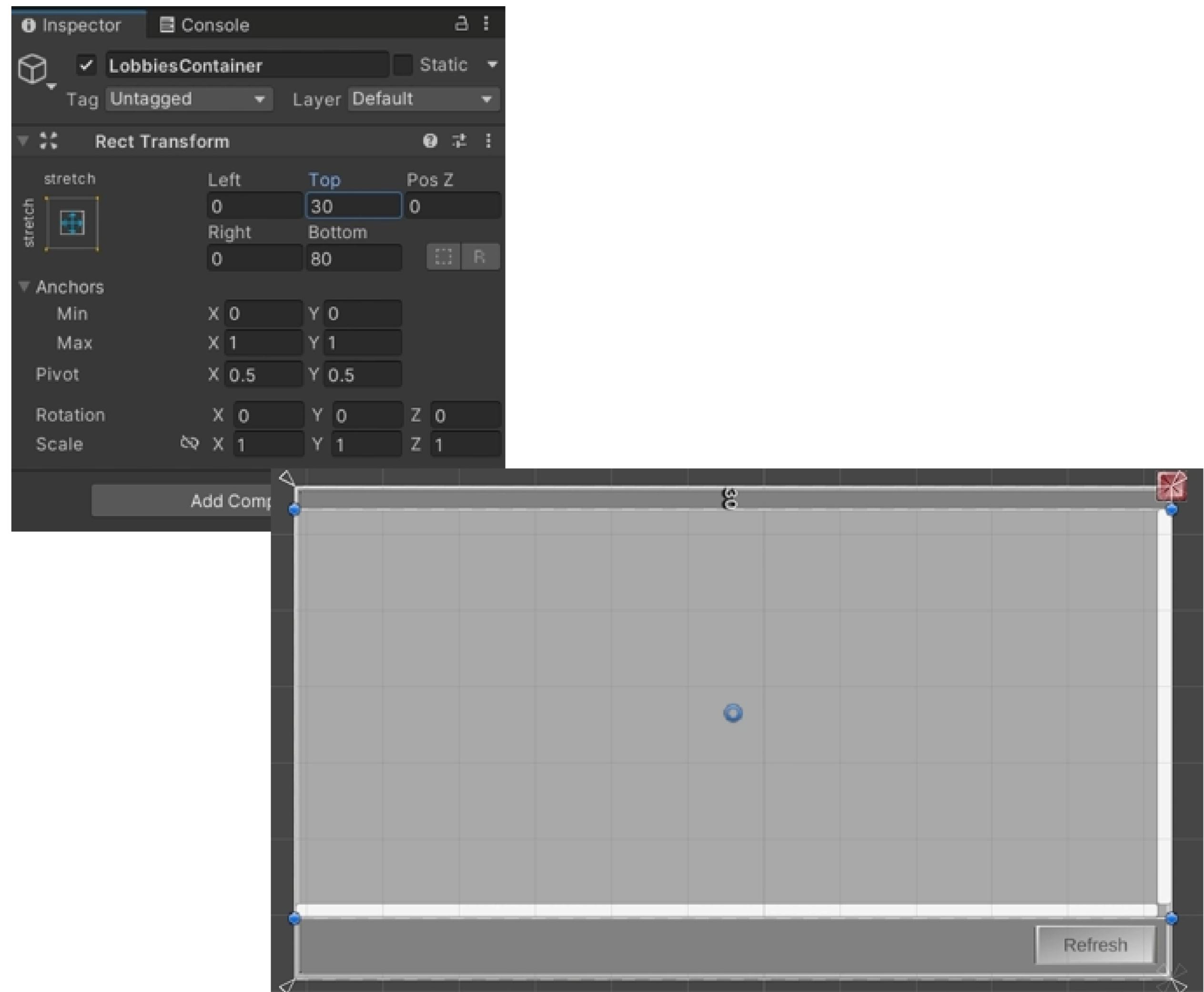
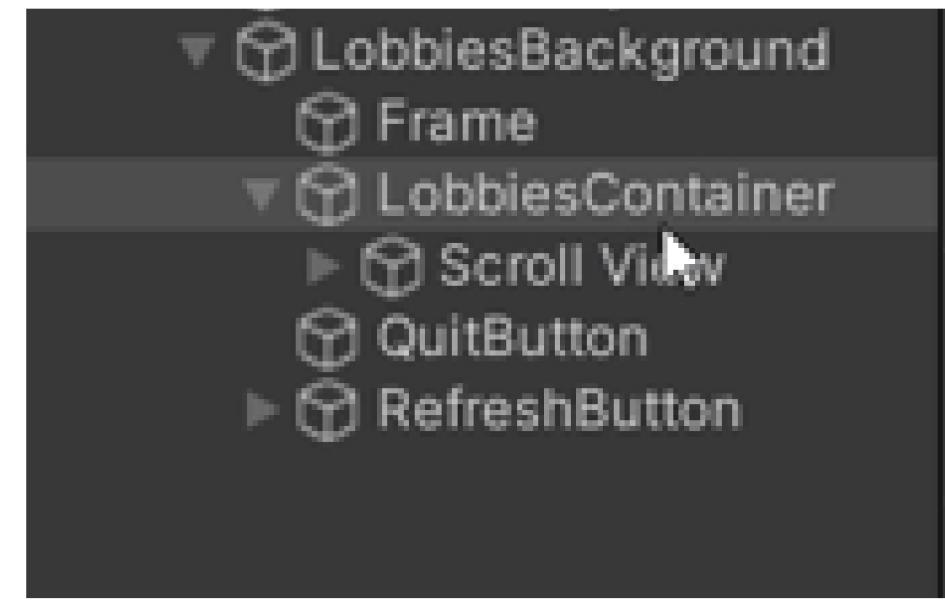
- Frame
- QuitButton
- RefreshButton
- ScrollView

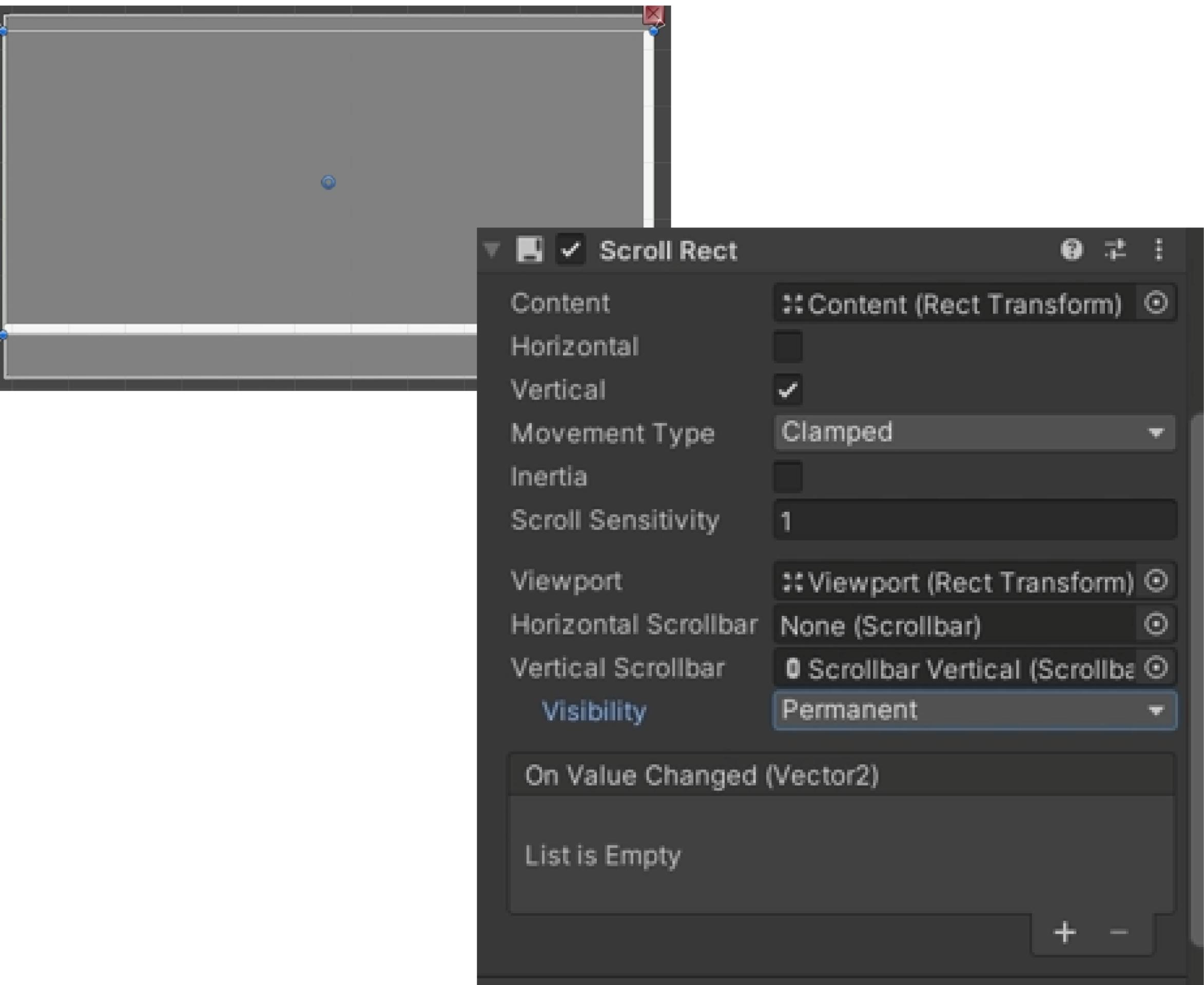
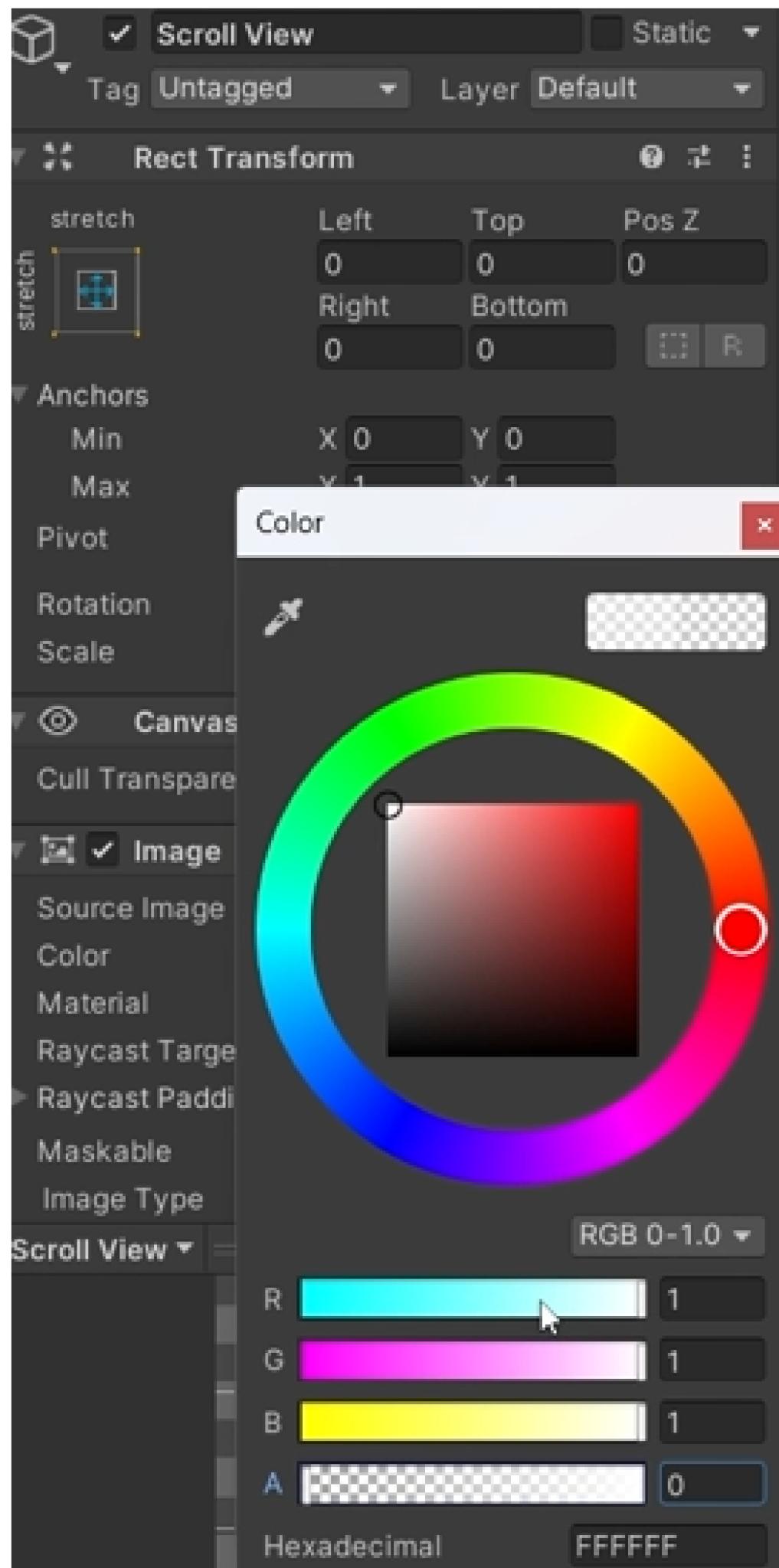
▼ LobbiesBackground

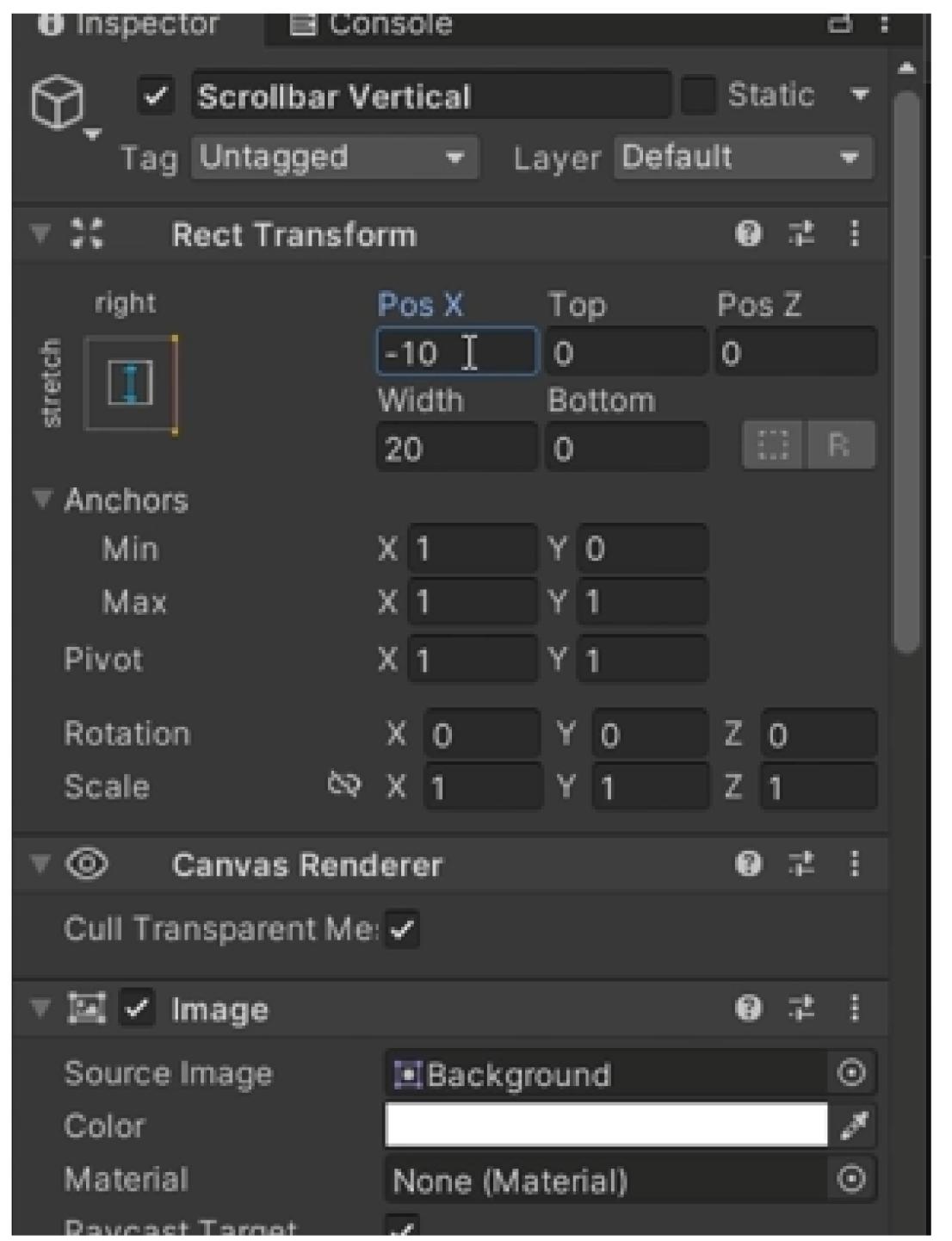
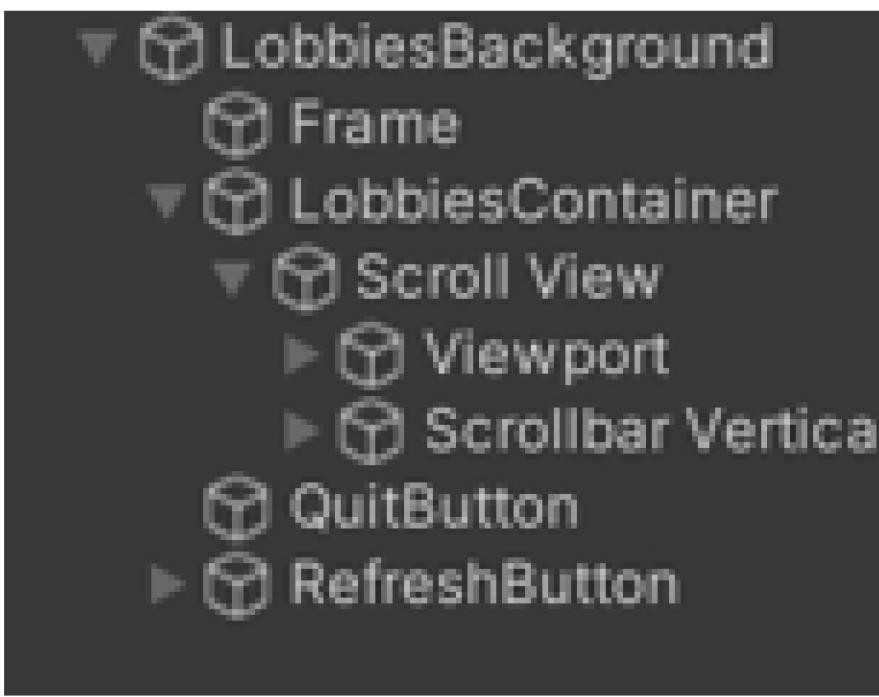
- Frame
- LobbiesContainer
- QuitButton
- RefreshButton
- ScrollView

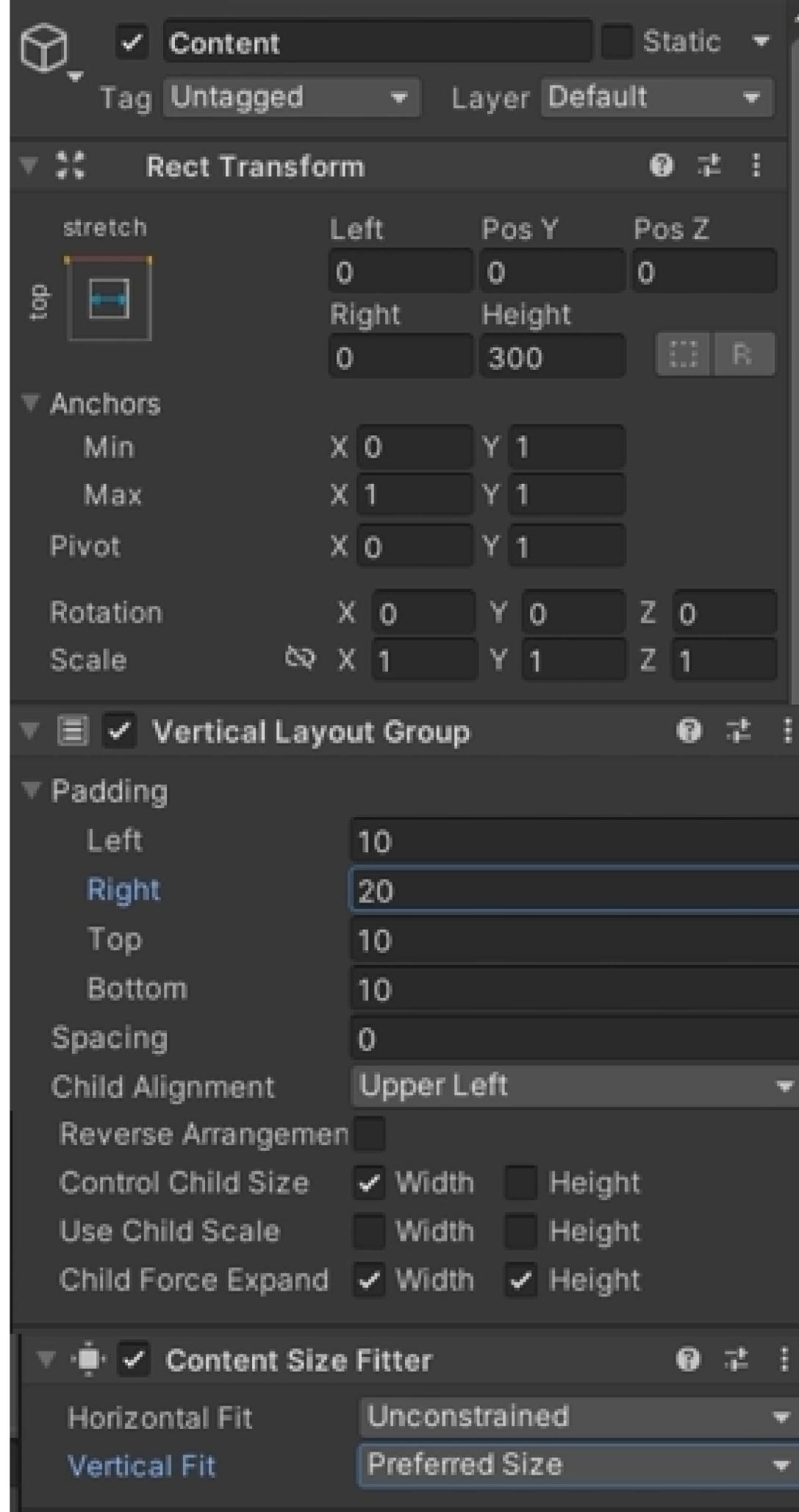
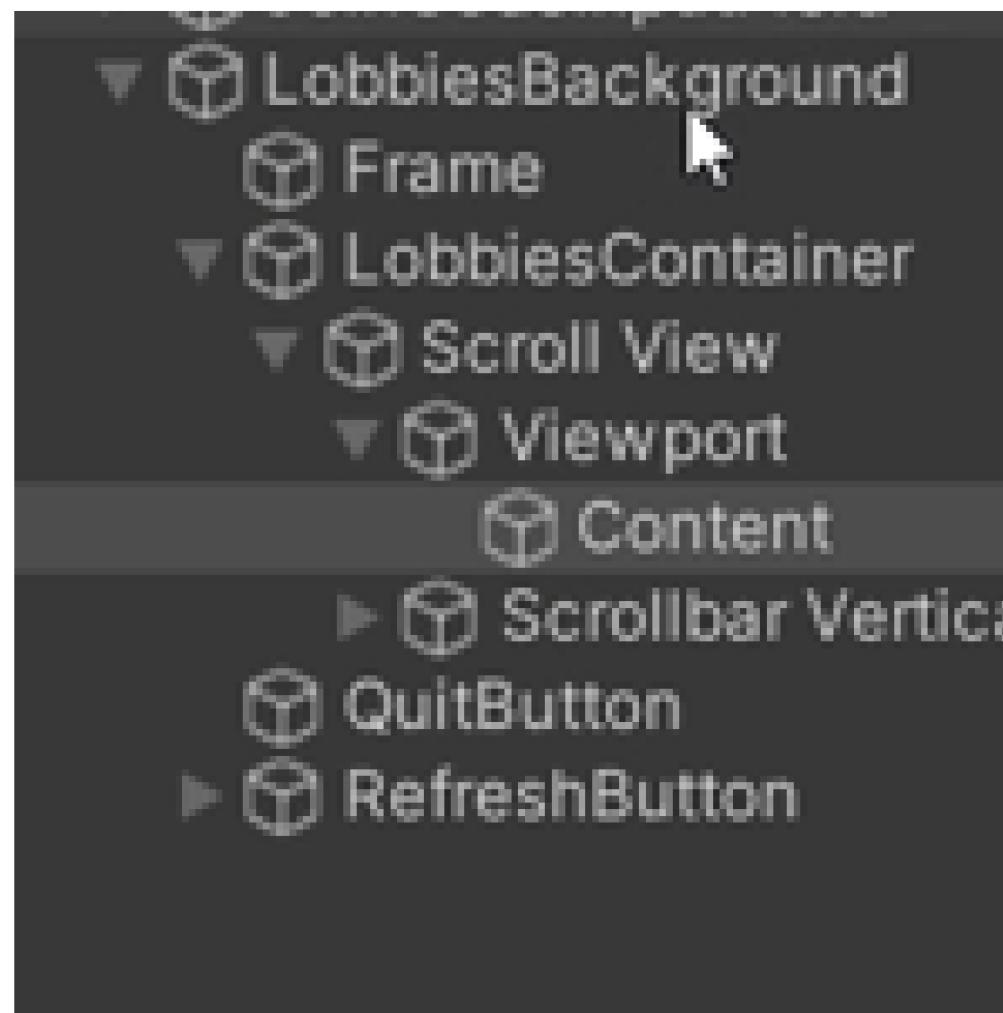


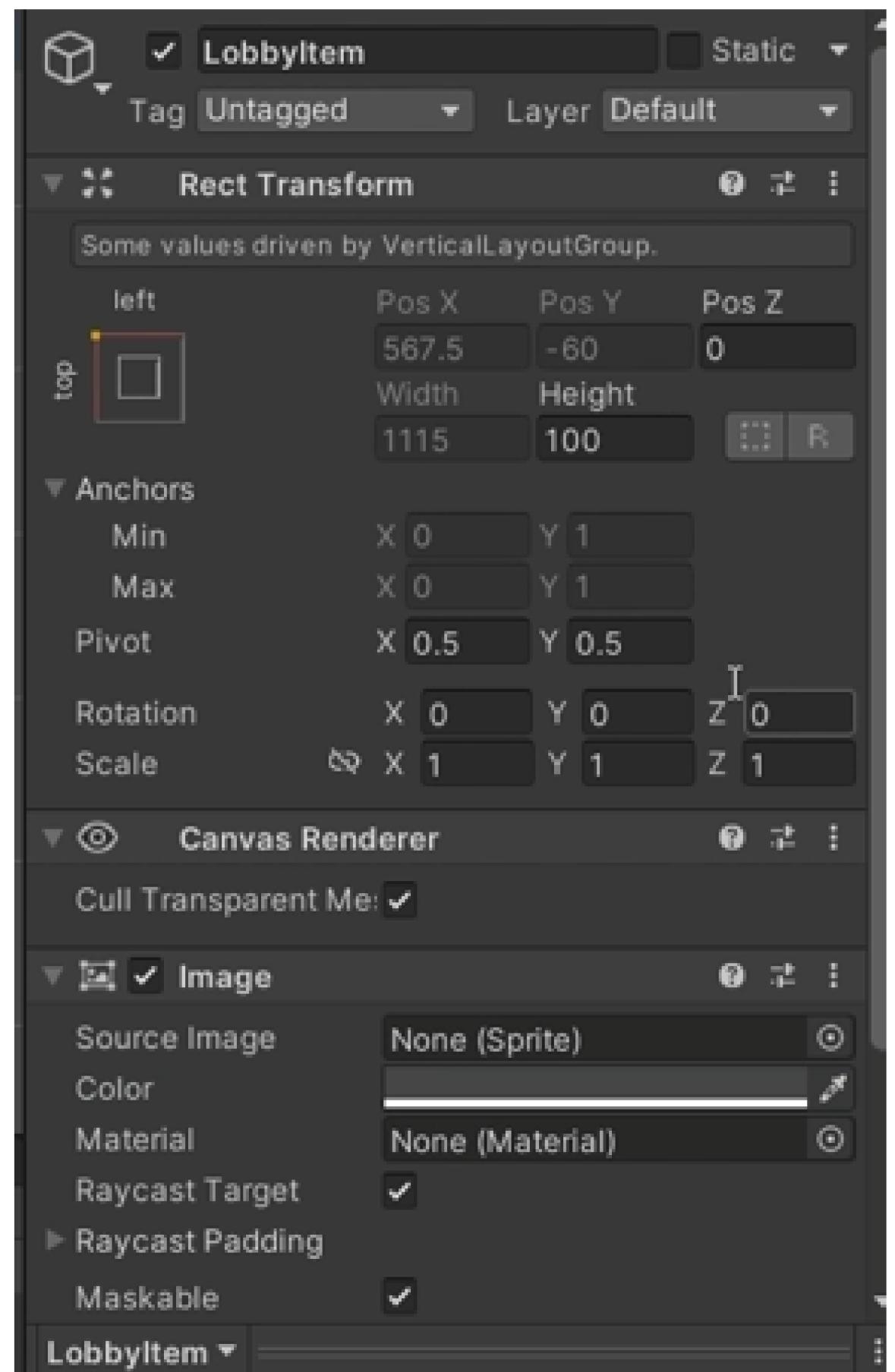








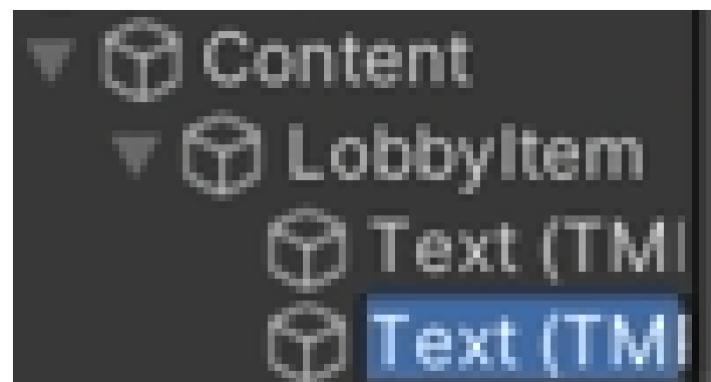
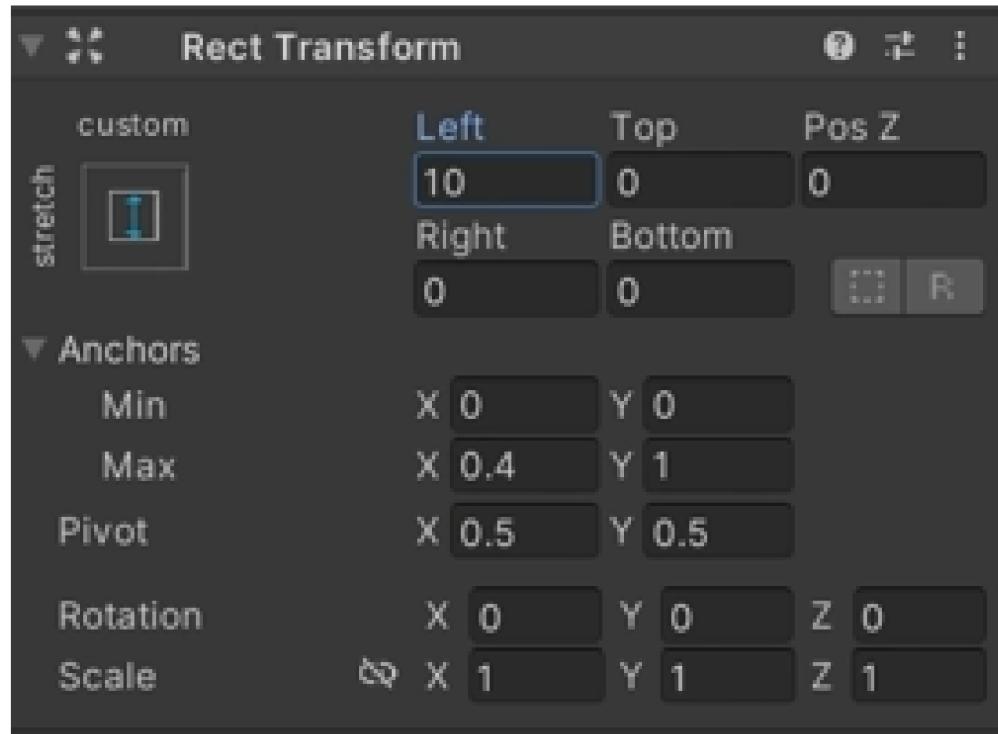




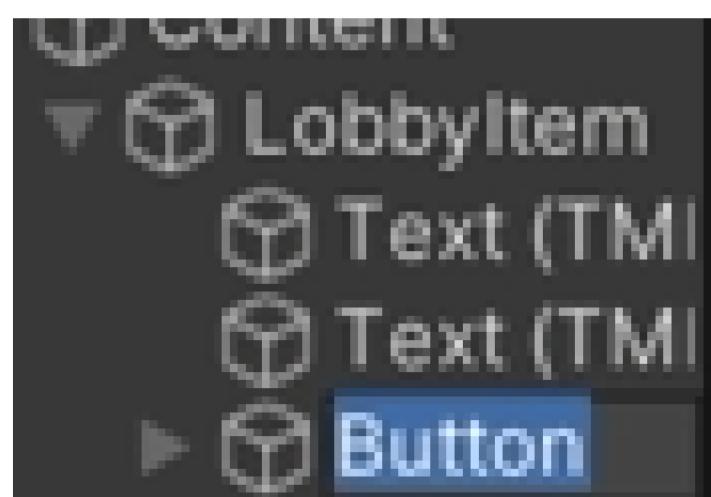
LobbyNameText



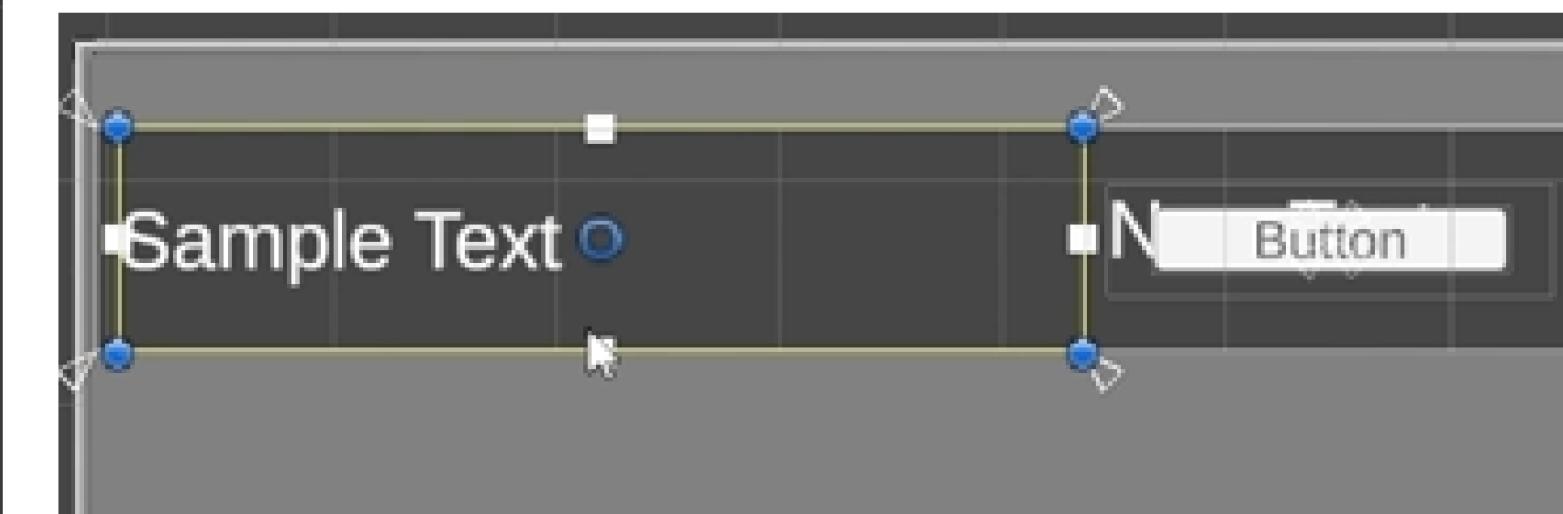
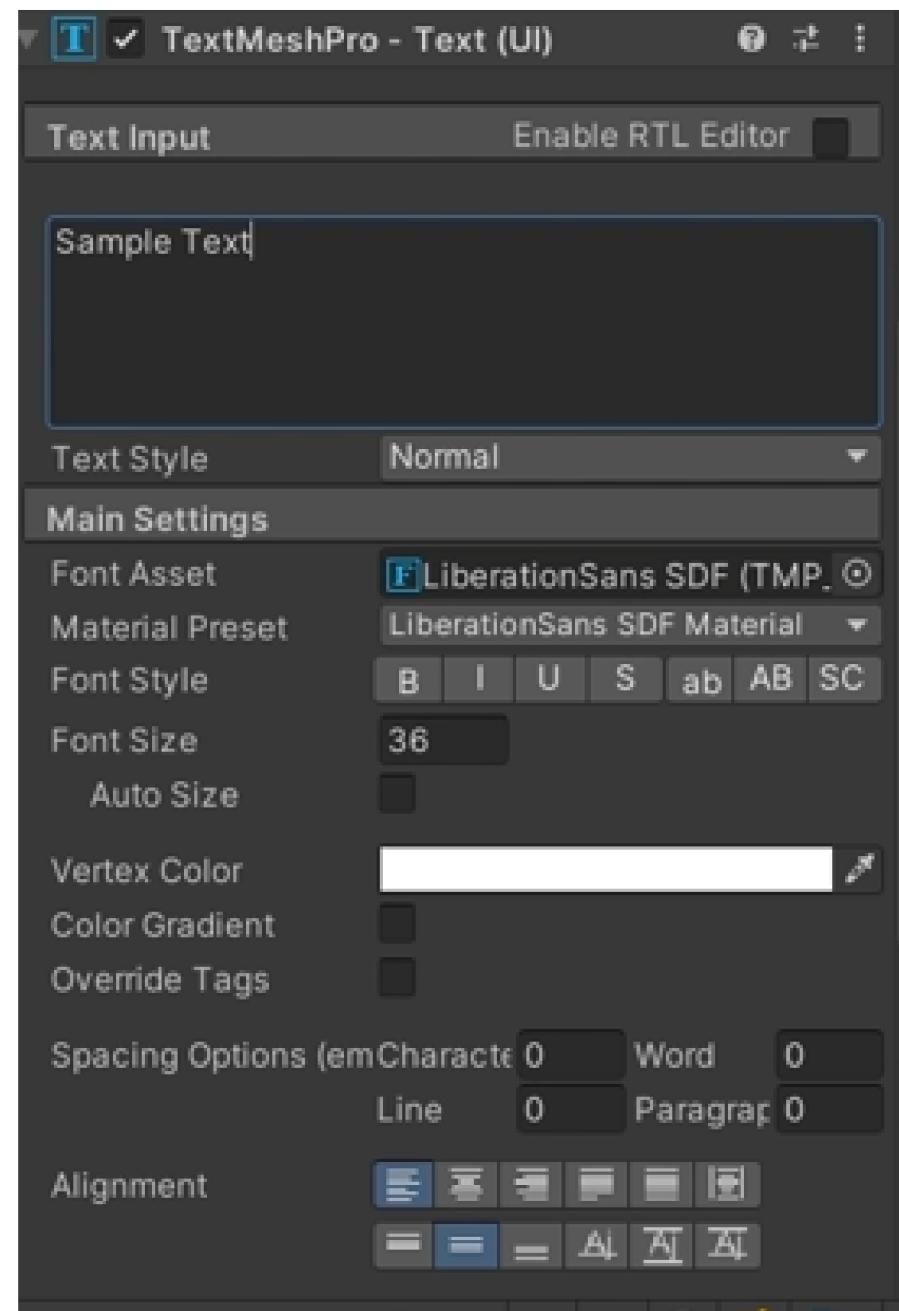
LobbyNameText



LobbyPlayersText



JoinButton



LobbyPlayersText



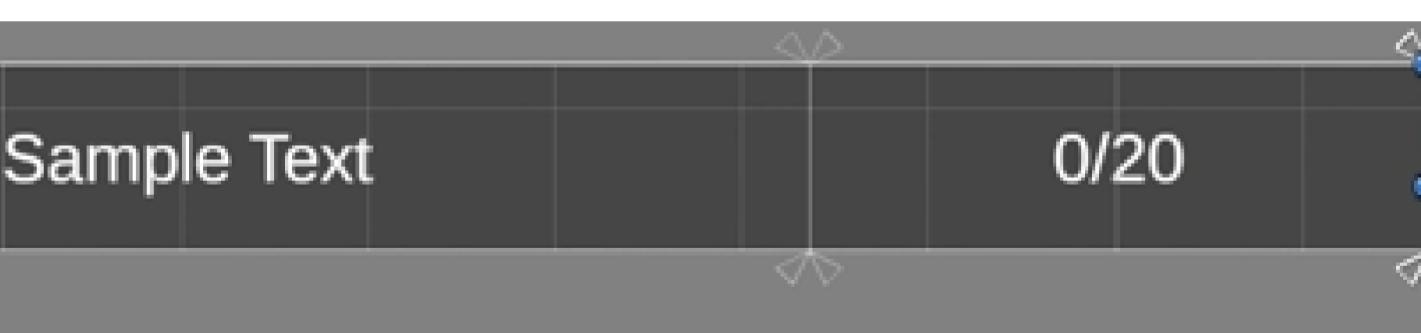
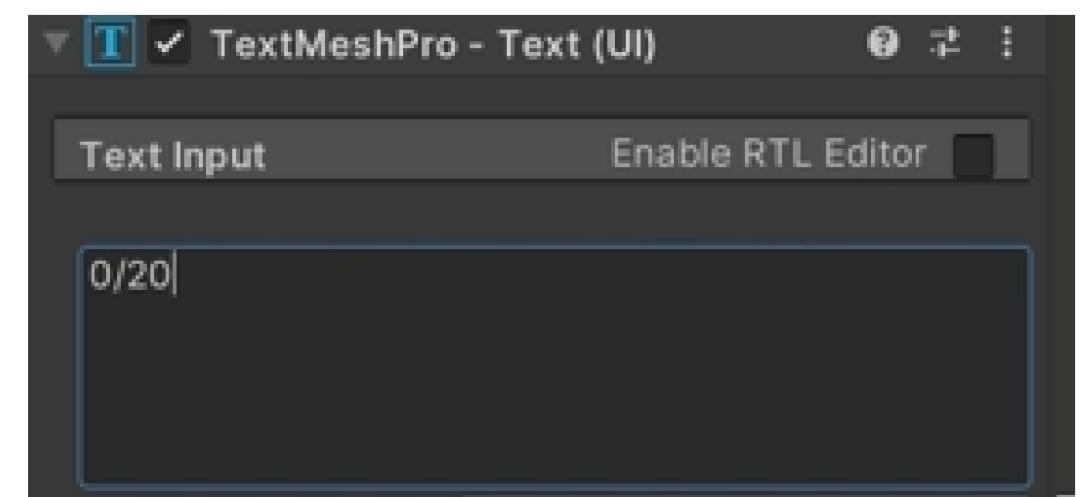
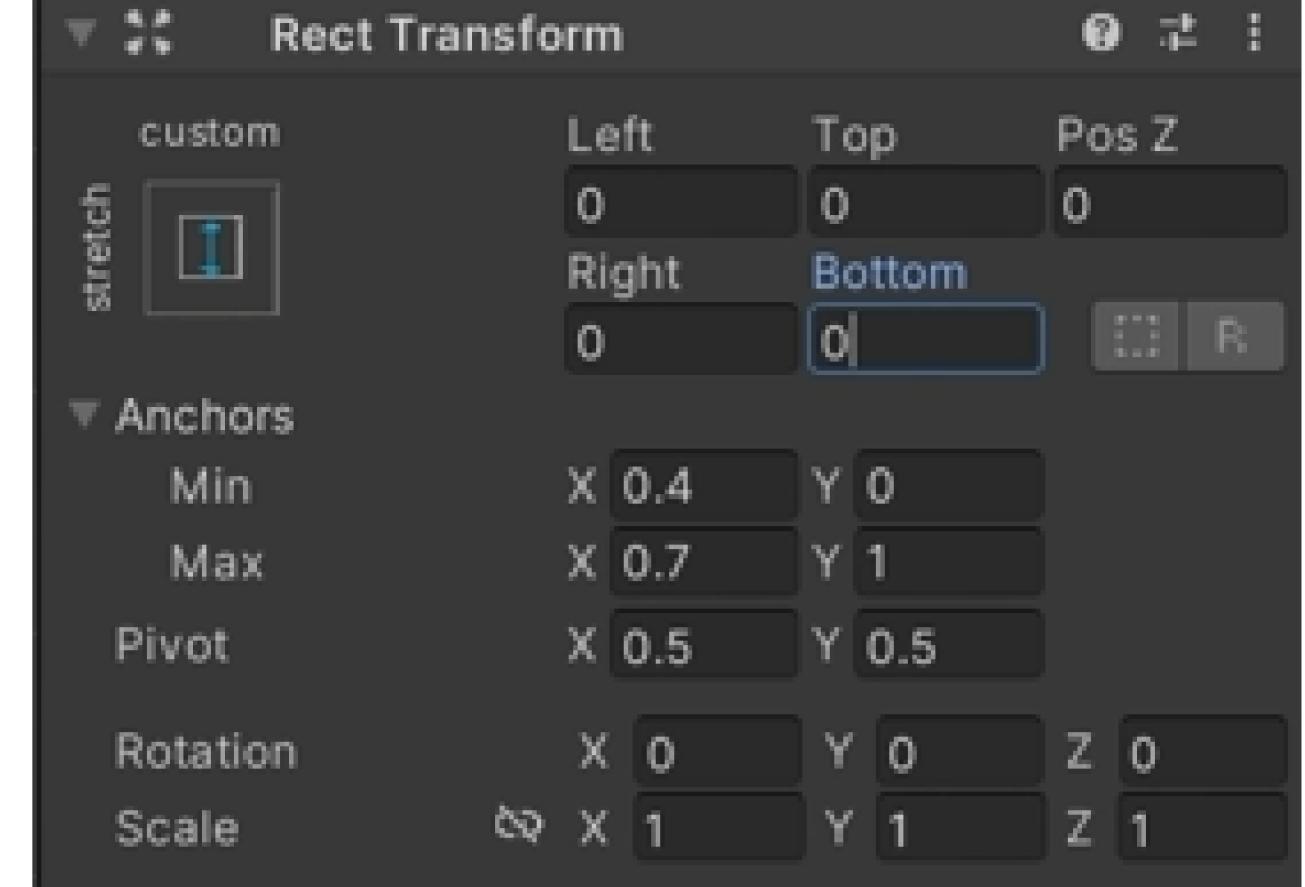
LobbyNameText



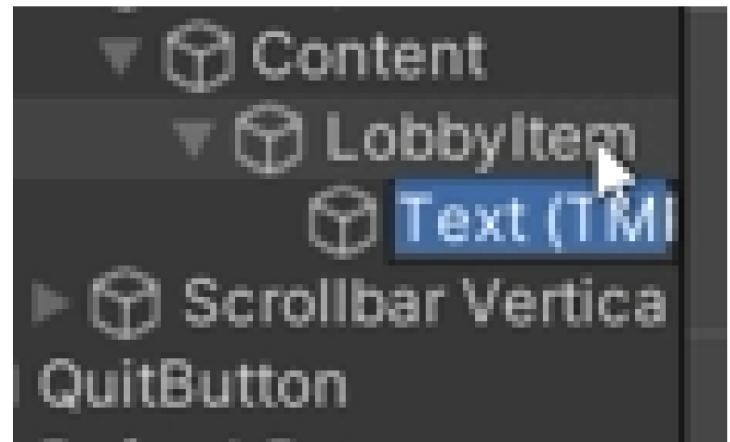
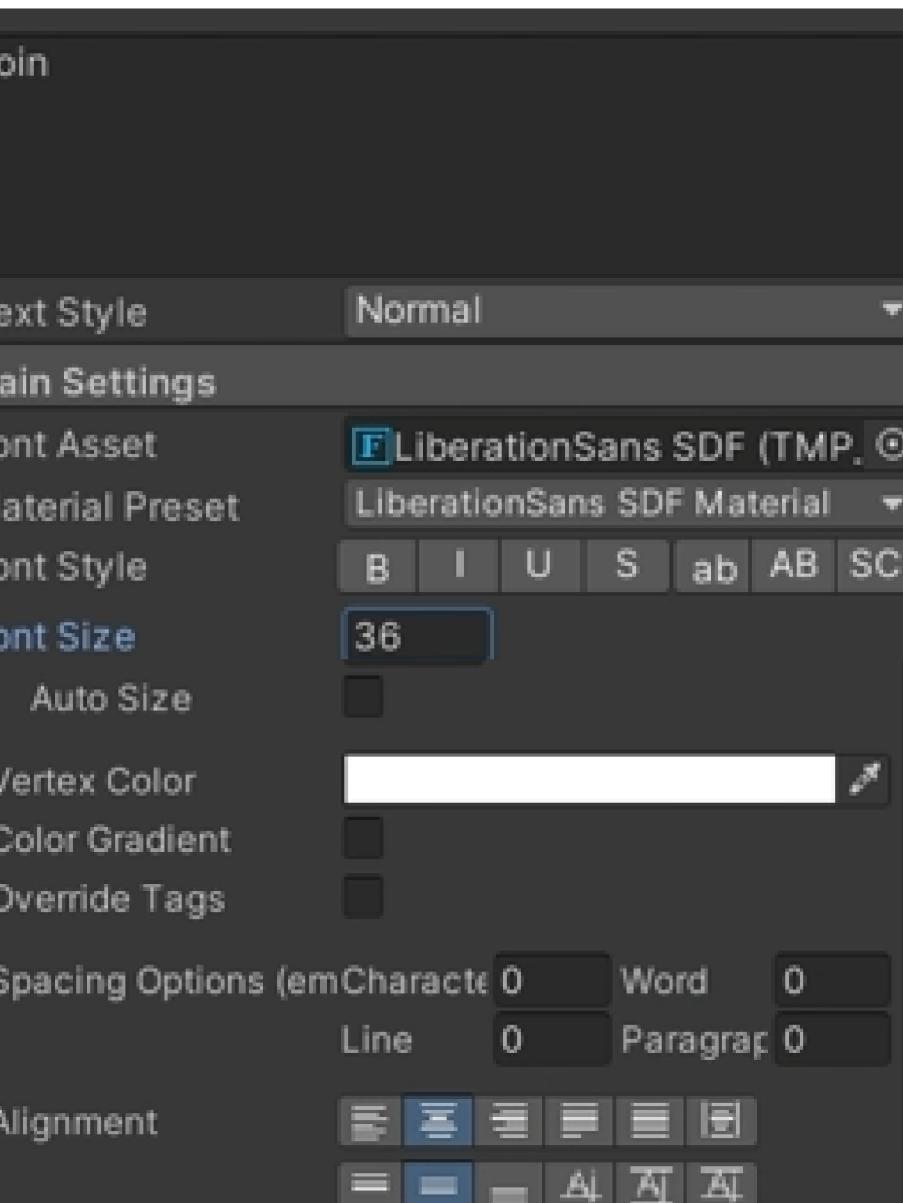
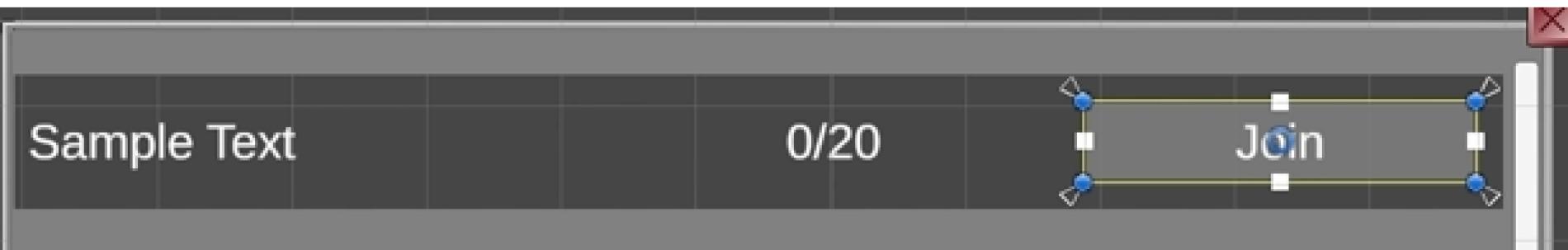
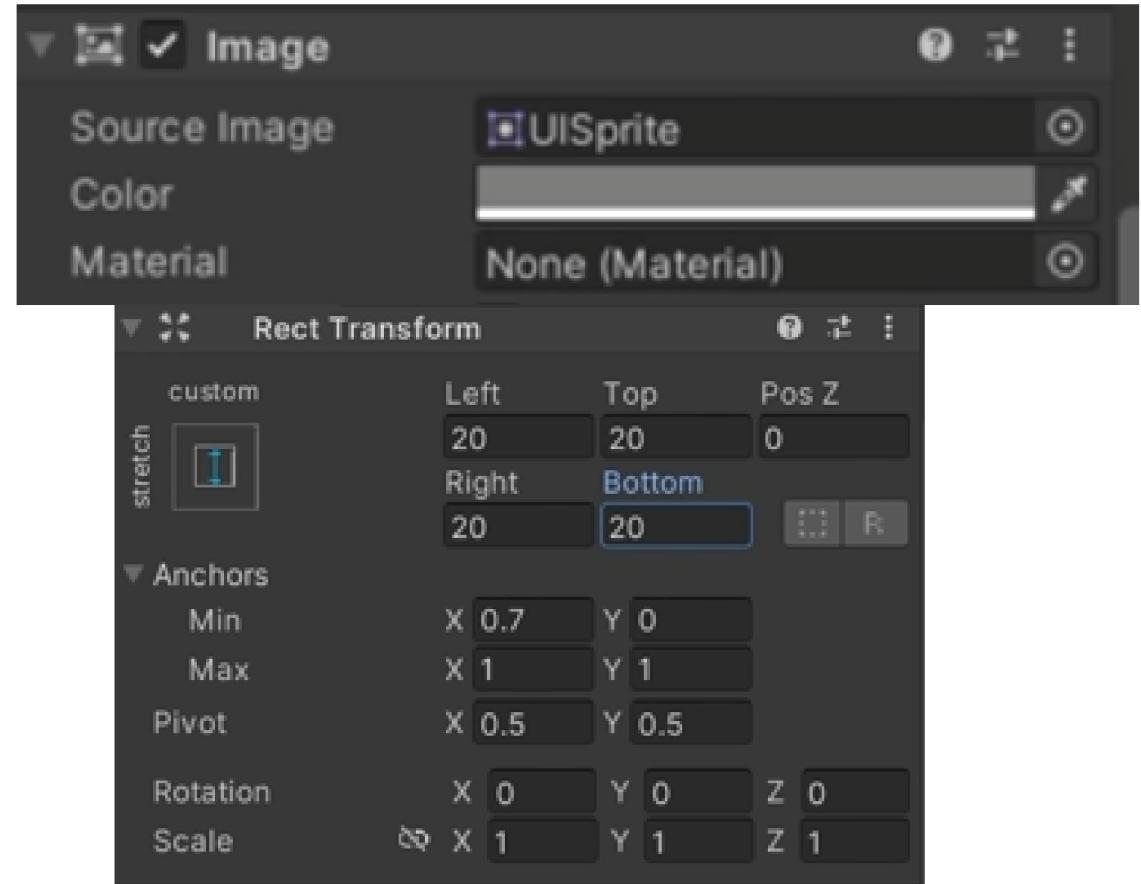
LobbyPlayersText



JoinButton



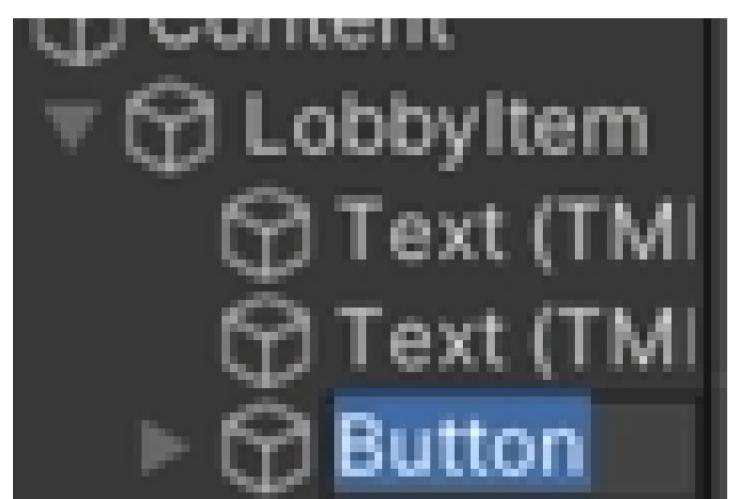
JoinButton



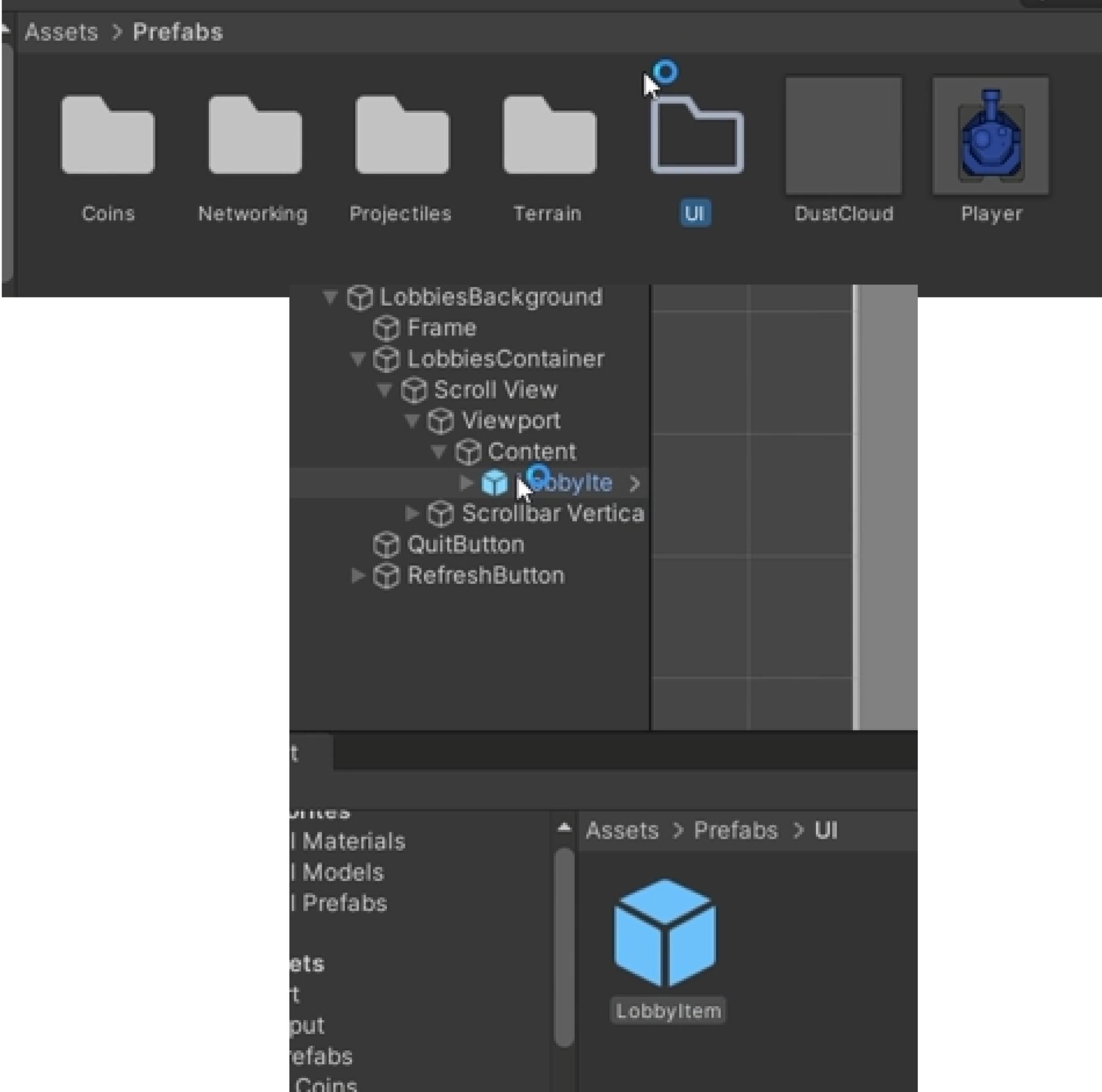
LobbyNameText

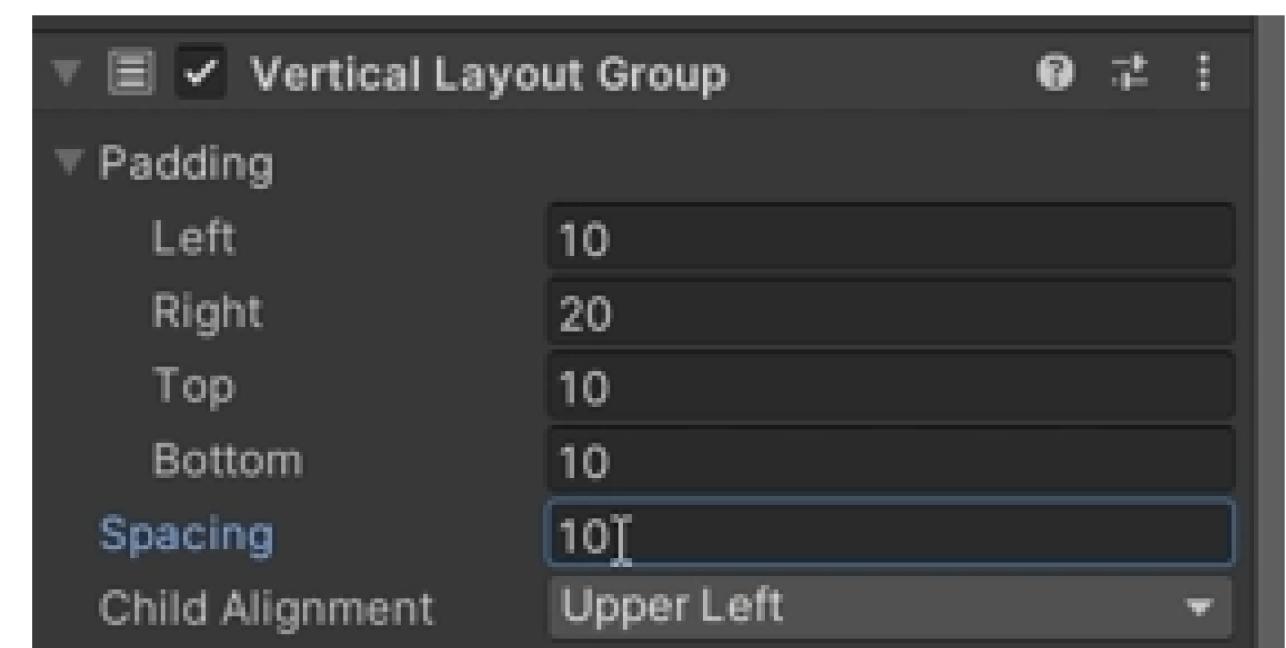
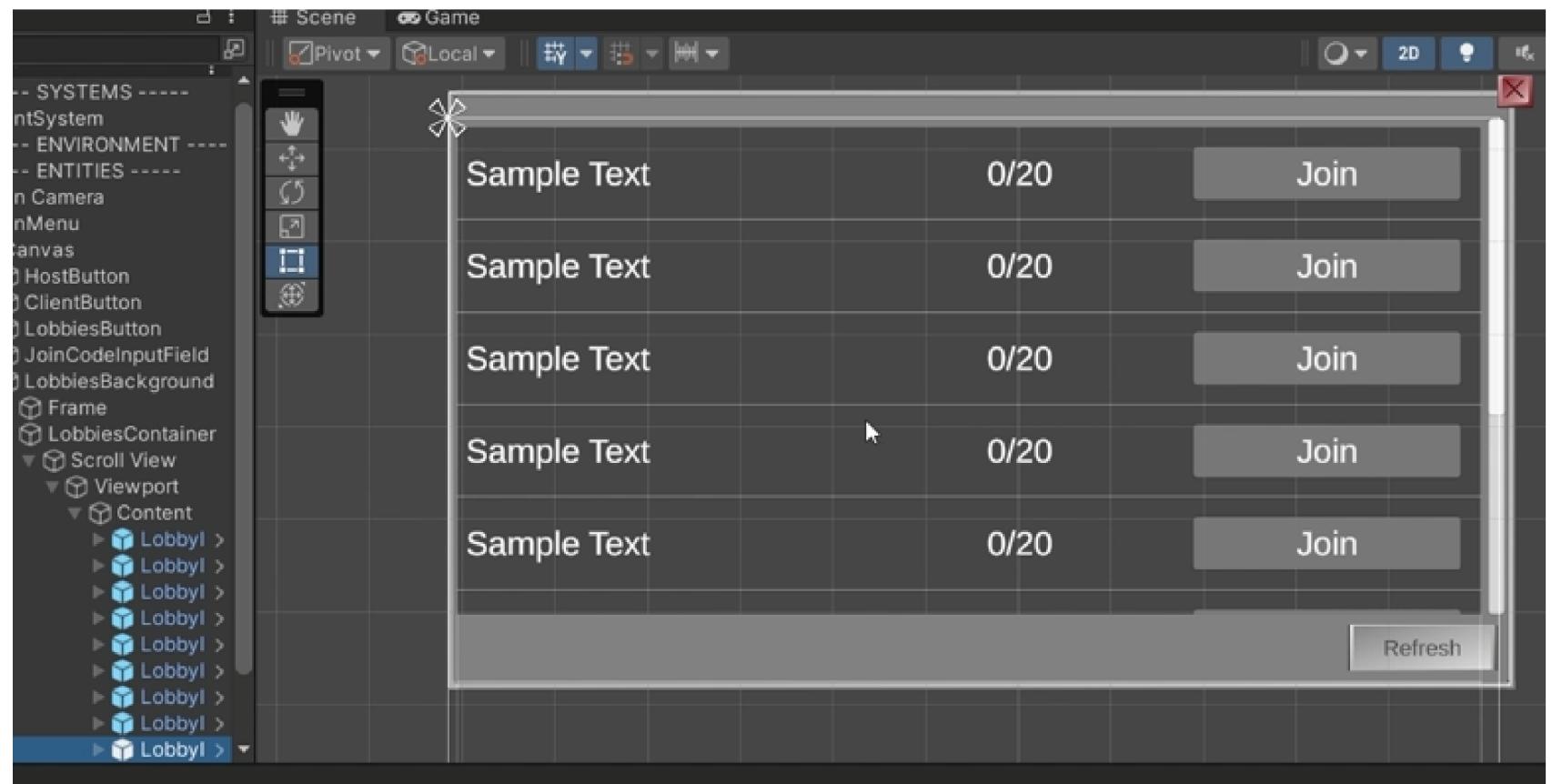


LobbyPlayersText



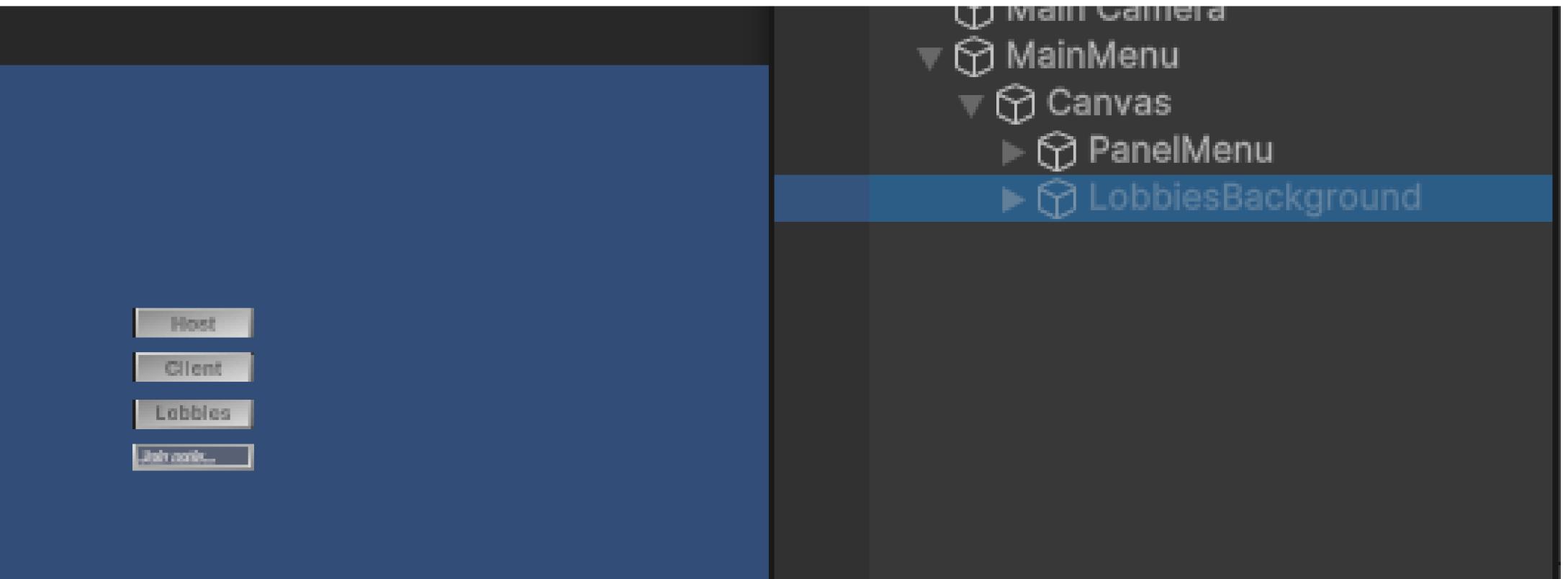
JoinButton





Creating Lobbies

```
▶ 🏠 LobbiesBackground
  ◀ 🏠 Frame
  ▶ 🏠 LobbiesContainer
    ▶ 🏠 Scroll View
      ▶ 🏠 Viewport
        ◀ 🏠 Content
      ▶ 🏠 Scrollbar Vertical
    ◀ 🏠 QuitButton
  ▶ 🏠 RefreshButton
```



มาที่โค้ด HostGameManager

```
41     }
42
43     UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
44
45     RelayServerData relayServerData = new ... RelayServerData(allocation, "dtls");
46     transport.SetRelayServerData(relayServerData);
47
48     NetworkManager.Singleton.StartHost();
49
50     NetworkManager.Singleton.SceneManager.LoadScene(GameSceneName, LoadSceneMode.Single);
51
52 }
```

```
try
{
    -
    -
}
catch (LobbyServiceException e)
{
    Debug.Log(e);
    return;
}
```

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Threading.Tasks;
using Unity.Netcode;
using Unity.Netcode.Transports.UTP;
using Unity.Networking.Transport.Relay;
using Unity.Services.Lobbies;
using Unity.Services.Lobbies.Models;
using Unity.Services.Relay;        try
using Unity.Services.Relay.Models; {
using UnityEngine;
using UnityEngine.SceneManagement;
```

```
CreateLobbyOptions lobbyOptions = new CreateLobbyOptions();
lobbyOptions.IsPrivate = false;
lobbyOptions.Data = new Dictionary<string, DataObject>()
{
    {
        "JoinCode", new DataObject(
            visibility: DataObject.VisibilityOptions.Member,
            value: joinCode
        )
    }
};
Lobby lobby = await Lobbies.Instance.CreateLobbyAsync(
    "My Lobby", MaxConnections, lobbyOptions);
}
```

```
2 references
public class HostGameManager
{
    private Allocation allocation;
    private string joinCode;
    private string lobbyId;

    private const int MaxConnections = 20;
    private const string GameSceneName = "Game";
    1 reference

    try
    {
        CreateLobbyOptions lobbyOptions = new CreateLobbyOptions();
        lobbyOptions.IsPrivate = false;
        lobbyOptions.Data = new Dictionary<string, DataObject>()
        {
            {
                "JoinCode", new DataObject(
                    visibility: DataObject.VisibilityOptions.Member,
                    value: joinCode
                )
            }
        };
        Lobby lobby = await Lobbies.Instance.CreateLobbyAsync(
            "My Lobby", MaxConnections, lobbyOptions);
        lobbyId = lobby.Id;
    }
}
```

Understanding and Implementing Lobby Heartbeats



Brandon Wofford

9 months ago · Updated

Follow

The Lobby service plays an essential role in setting up multiplayer gaming sessions. One important aspect of managing lobbies is maintaining their active state, which is achieved using the concept of "heartbeats". This guide will take you through the basics of lobby heartbeats, why they are crucial, and how to implement them in your Unity project.

Why do we need Lobby Heartbeats?

Lobby is built on top of stateless networking protocols, meaning it doesn't inherently know whether a connection is still alive or not. To determine if a lobby host is still online, we employ heartbeats, where the host sends an occasional "keep-alive" message to the service.

If the Lobby service doesn't receive a heartbeat from a host within 30 seconds, the lobby is marked as inactive and won't appear in query results. The lobby can be reactivated if a heartbeat is received within an hour. However, if a lobby fails to send a heartbeat for over an hour, it's considered expired. It can't be reactivated and is automatically deleted by the service.

This system is vital for maintaining efficient networking, as it helps to prevent players from continually finding and attempting to join abandoned lobbies.

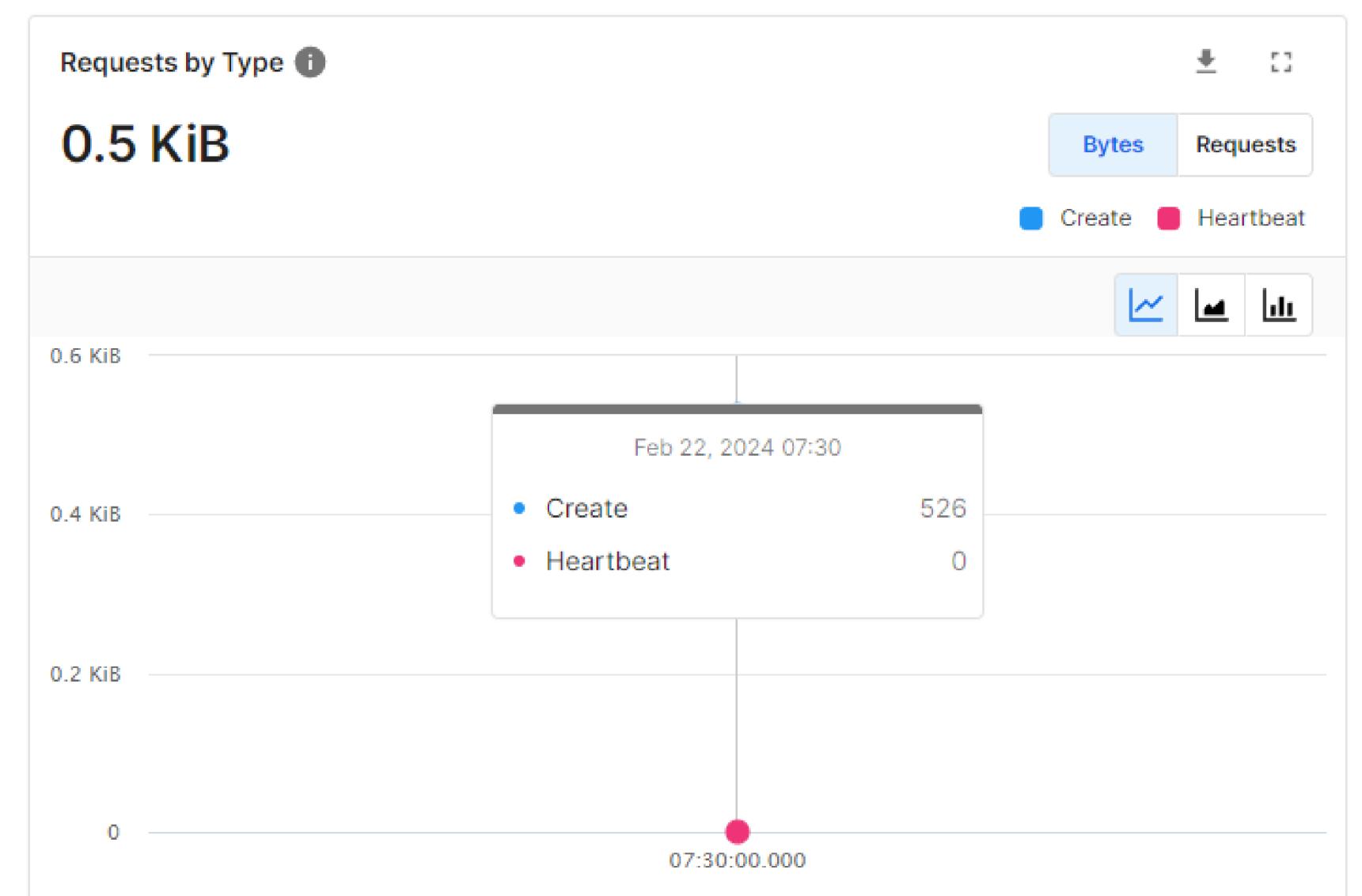
Heartbeating a Lobby in Unity

To heartbeat a lobby in Unity, you need to periodically send a heartbeat request to the Lobby service. Below is an example of how you can create a new lobby and integrate a coroutine to send a heartbeat every 15 seconds.

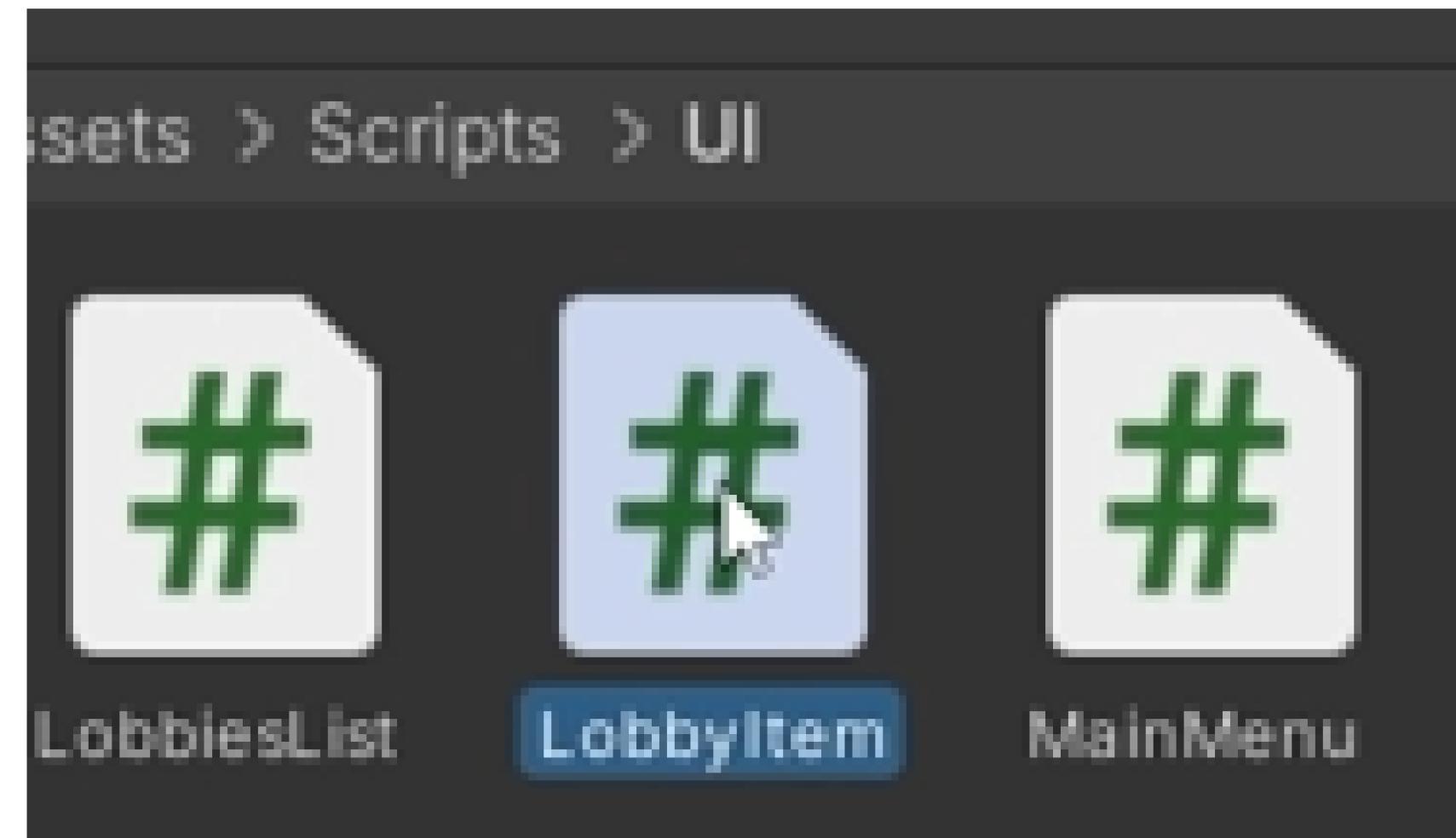
```
68
69     HostSingleton.Instance.StartCoroutine(HeartbeatLobby(15));
70 }
71 catch (LobbyServiceException e)
72 {
73     Debug.Log(e);
74     return;
75 }
76
77 NetworkManager.Singleton.StartHost();
78
79 NetworkManager.Singleton.SceneManager.LoadScene(GameSceneName, LoadSceneMode.Single);
80 }
81
82 private IEnumerator HeartbeatLobby(float waitTimeSeconds)
83 {
84     yield return null;
85 }
86
87
```

1 reference

```
private IEnumerator HeartbeatLobby(float waitTimeSeconds)
{
    WaitForSecondsRealtime delay = new WaitForSecondsRealtime(waitTimeSeconds);
    while(true)
    {
        Lobbies.Instance.SendHeartbeatPingAsync(lobbyId);
        yield return delay;
    }
}
```



Joining Lobbies



```
1  [-]using System.Collections;
2  [-]using System.Collections.Generic;
3  [-]using TMPro;
4  [-]using Unity.Services.Lobbies.Models;
5  [-]using UnityEngine;
6
7  [-]public class LobbyItem : MonoBehaviour
8  {
9  [-]    [SerializeField] private TMP_Text lobbyNameText;
10 [-]   [SerializeField] private TMP_Text lobbyPlayerText;
```

```
Unity Script | 0 references
7  -public class LobbyItem : MonoBehaviour
8  {
9      [SerializeField] private TMP_Text lobbyNameText;
10     [SerializeField] private TMP_Text lobbyPlayerText;
11
12     private LobbiesList lobbiesList;
13     0 references
14     public void Initialise(LobbiesList lobbiesList, Lobby lobby)
15     {
16         this.lobbiesList = lobbiesList;
17         lobbyNameText.text = lobby.Name;
18         lobbyPlayerText.text = $"{lobby.Players.Count}/{lobby.MaxPlayers}";
19     }
20 }
```

```
④ Unity Script | 0 references
7  [-]public class LobbyItem : MonoBehaviour
8  {
9      [SerializeField] private TMP_Text lobbyNameText;
10     [SerializeField] private TMP_Text lobbyPlayerText;
11
12     private LobbiesList lobbiesList;
13     private Lobby lobby;
14     0 references
15     [-]public void Initalise(LobbiesList lobbiesList,Lobby lobby)
16     {
17         this.lobbiesList = lobbiesList;
18         this.lobby = lobby;
19         lobbyNameText.text = lobby.Name;
20         lobbyPlayerText.text = $"{lobby.Players.Count}/{lobby.MaxPlayers}";
21     }
22     0 references
23     [-]public void Join()
24     {
25         lobbiesList.JoinAsync(lobby);
26     }
}
```

Join Lobby

- **public async void JoinAsync(Lobby)**
- Create a try-catch block
- Make the call to

Lobbies.Instance.JoinLobbyByIdAsync

- Store the return value in a variable

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Unity.Services.Lobbies;
4  using Unity.Services.Lobbies.Models;
5  using UnityEngine;
6
7  public class LobbiesList : MonoBehaviour
8  {
9      private bool isJoining;
10     public async void JoinAsync(Lobby lobby)
11     {
12         if(isJoining) { return; }
13         isJoining = true;
14         try
15         {
16             Lobby joiningLobby = await Lobbies.Instance.JoinLobbyByIdAsync(lobby.Id);
17             string joinCode = joiningLobby.Data["JoinCode"].Value;
18
19             await ClientSingleton.Instance.GameManager.StartClientAsync(joinCode);
20         }
21         catch(LobbyServiceException e)
22         {
23             Debug.Log(e);
24         }
25         isJoining = false;
26     }
27 }
28
```

```
Unity Script (1 asset reference) | 2 references
public class LobbiesList : MonoBehaviour
{
    [SerializeField] private Transform lobbyItemParent;
    [SerializeField] private LobbyItem lobbyItemPrefab;
    private bool isJoining;
    private bool isRefreshing;
```

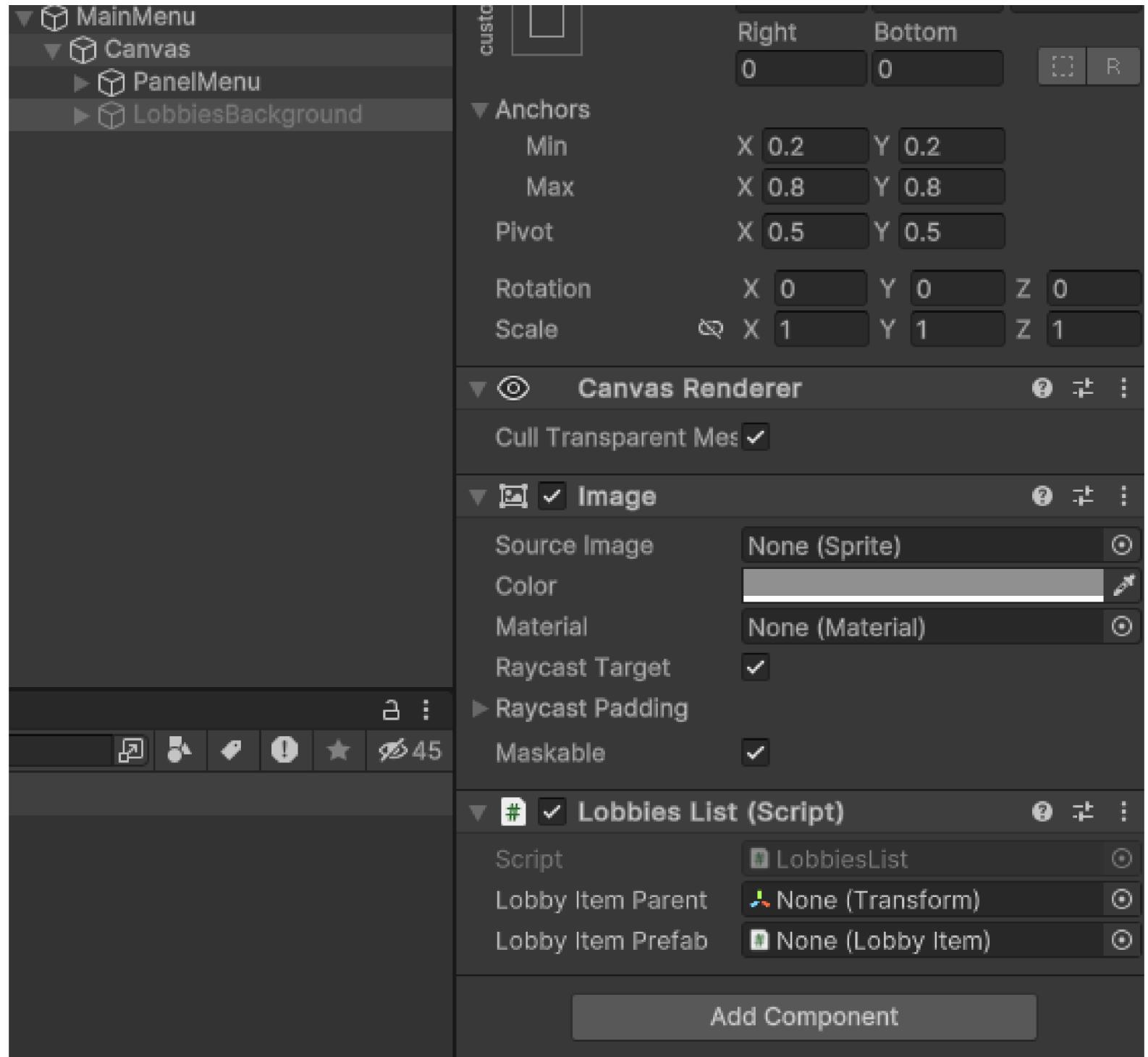
```
LobbiesList.cs* + X LobbyItem.cs ClientSingleton.cs MainMenu.cs ApplicationController.cs Cli
Assembly-CSharp
  LobbiesList
10 [SerializeField] private LobbyItem lobbyItemPrefab;
11
12 private bool isJoining;
13 private bool isRefreshing;
14 Unity Message | 0 references
15 private void OnEnable()
16 {
17     RefreshList();
18 }
19 1 reference
20 public async void RefreshList()
21 {
22     if (isRefreshing) { return; }
23     isRefreshing = true;
24     try
25     {
26         QueryLobbiesOptions options = new QueryLobbiesOptions();
27         options.Count = 25;
28         options.Filters = new List<QueryFilter>()
29         {
30             new QueryFilter(
31                 field: QueryFilter.FieldOptions.AvailableSlots,
32                 op: QueryFilter.OpOptions.GT,
33                 value: "0"),
34             new QueryFilter(
35                 field: QueryFilter.FieldOptions.IsLocked,
36                 op: QueryFilter.OpOptions.EQ,
37                 value: "0")
38         };
39
40         QueryResponse lobbies = await Lobbies.Instance.QueryLobbiesAsync(options);
41
42         foreach(Transform child in lobbyItemParent)
43         {
44             Destroy(child.gameObject);
45         }
46         foreach (Lobby lobby in lobbies.Results)
47         {
48             LobbyItem lobbyItem = Instantiate(lobbyItemPrefab, lobbyItemParent);
49             lobbyItem.Initalise(this, lobby);
50         }
51     catch (LobbyServiceException e)
52     {
53         Debug.Log(e);
54     }
55     isRefreshing = false;
56 }
```

```
private bool isRefreshing;
Unity Message | 0 references
private void OnEnable()
{
    RefreshList();
}

1 reference
public async void RefreshList()
{
    if (isRefreshing) { return; }
    isRefreshing = true;
    try
    {
        QueryLobbiesOptions options = new QueryLobbiesOptions();
        options.Count = 25;
        options.Filters = new List<QueryFilter>()
        {
            new QueryFilter(
                field: QueryFilter.FieldOptions.AvailableSlots,
                op: QueryFilter.OpOptions.GT,
                value: "0"),
            new QueryFilter(
                field: QueryFilter.FieldOptions.IsLocked,
                op: QueryFilter.OpOptions.EQ,
                value: "0")
        };
        QueryResponse lobbies = await Lobbies.Instance.QueryLobbiesAsync(options);

        foreach(Transform child in lobbyItemParent)
        {
            Destroy(child.gameObject);
        }
        foreach (Lobby lobby in lobbies.Results)
        {
            LobbyItem lobbyItem = Instantiate(lobbyItemPrefab, lobbyItemParent);
            lobbyItem.Initalise(this, lobby);
        }
    catch (LobbyServiceException e)
    {
        Debug.Log(e);
    }
    isRefreshing = false;
}
```

ໃຈ່ Menu Scene





Lobby Item (Script)

Script: LobbyItem

Lobby Name Text: LobbyNameText (Text Mesh)

Lobby Players Text: LobbyPlayersText (Text Mesh)

Default UI Material (Material): UI/Default

Add Component

LobbyItem

Canvas (Environment)

LobbyItem

LobbyNameText

LobbyPlayersText

JoinButton

JoinText

LobbyItem

Canvas (Environment)

LobbyItem

LobbyNameText

LobbyPlayersText

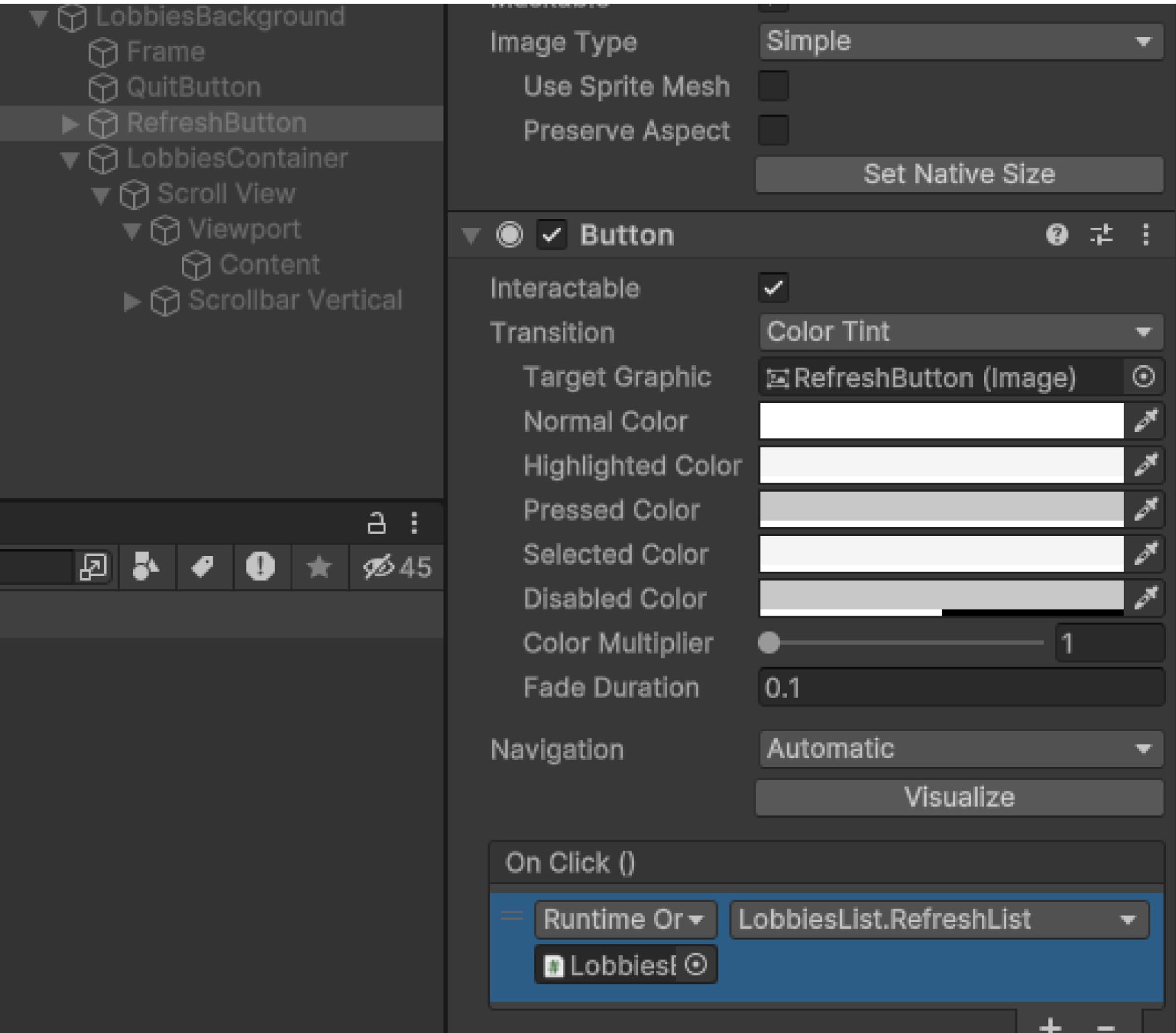
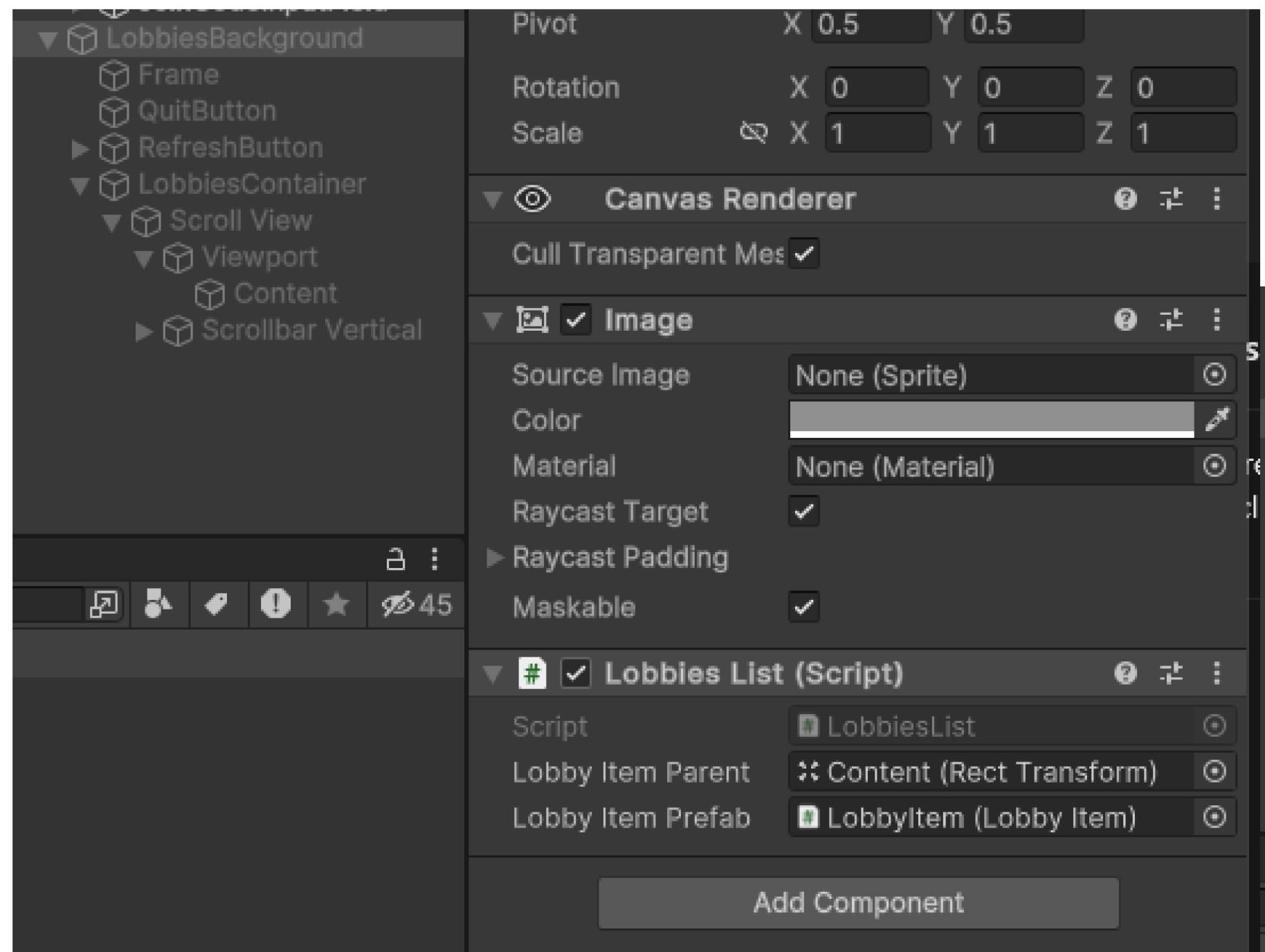
JoinButton

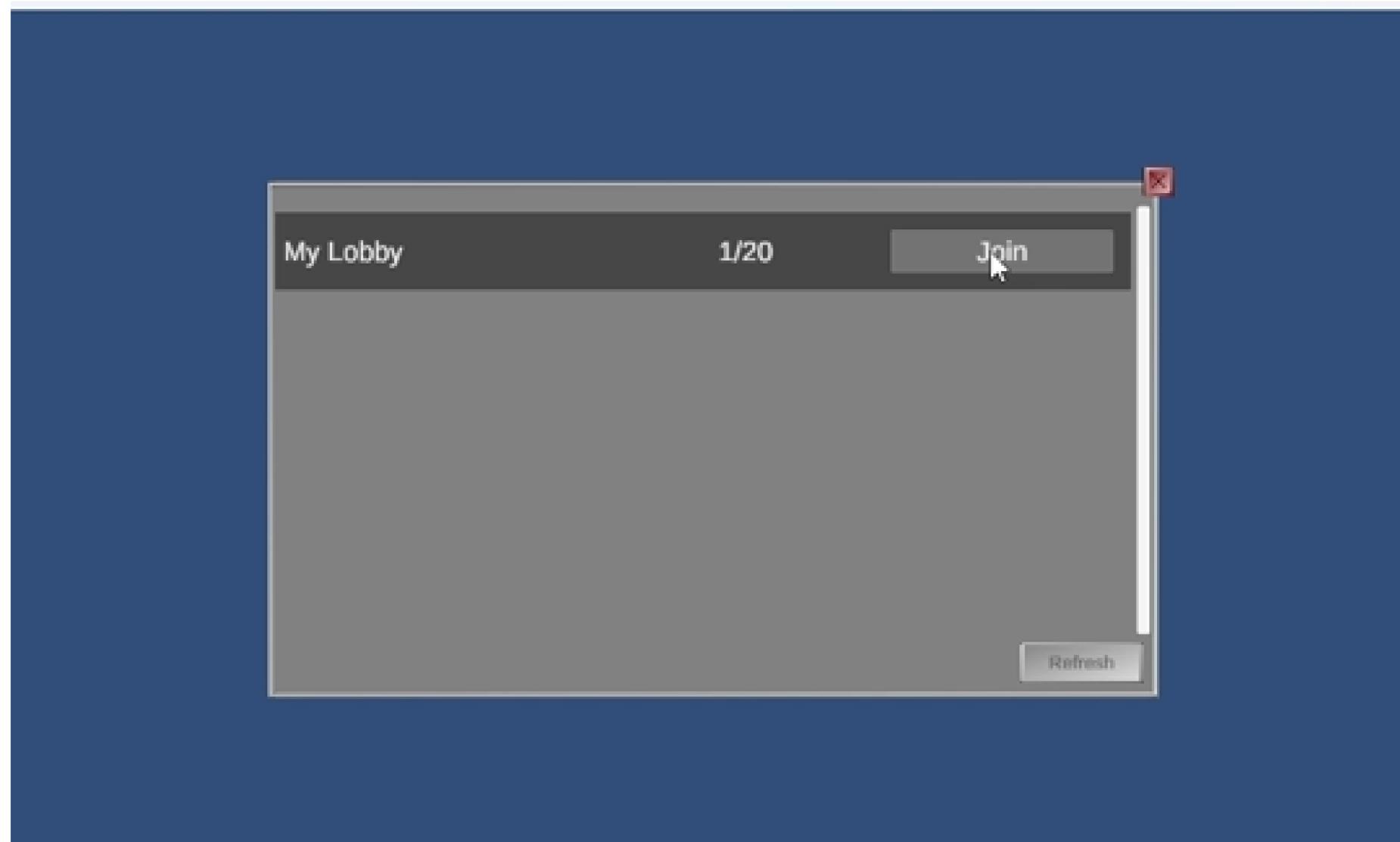
JoinText

On Click ()

Runtime Or LobbyItem.Join

LobbyItem





Assignment

ให้ทำการถ่ายคลิปผลลัพธ์ของการทำ Workshop ตามเนื้อหา Workshop ໃນແຕ່ລະ
หัวข้อนี้พร้อมອธิบายประกอบ

- Relay Service Setup
- Allocating A Relay
- Joining A Relay
- Lobbies UI
- Creating Lobbies
- Joining Lobbies



<https://discord.gg/24xmUFHzR>

WOLVEDEN ACADEMY



facebook.com/GameDevMew