

# 포톤네트워드를 사용한 멀티 플레이 게임 제작

# 포톤네트워킹을 사용한 FPS 개발

- 포톤네트워킹

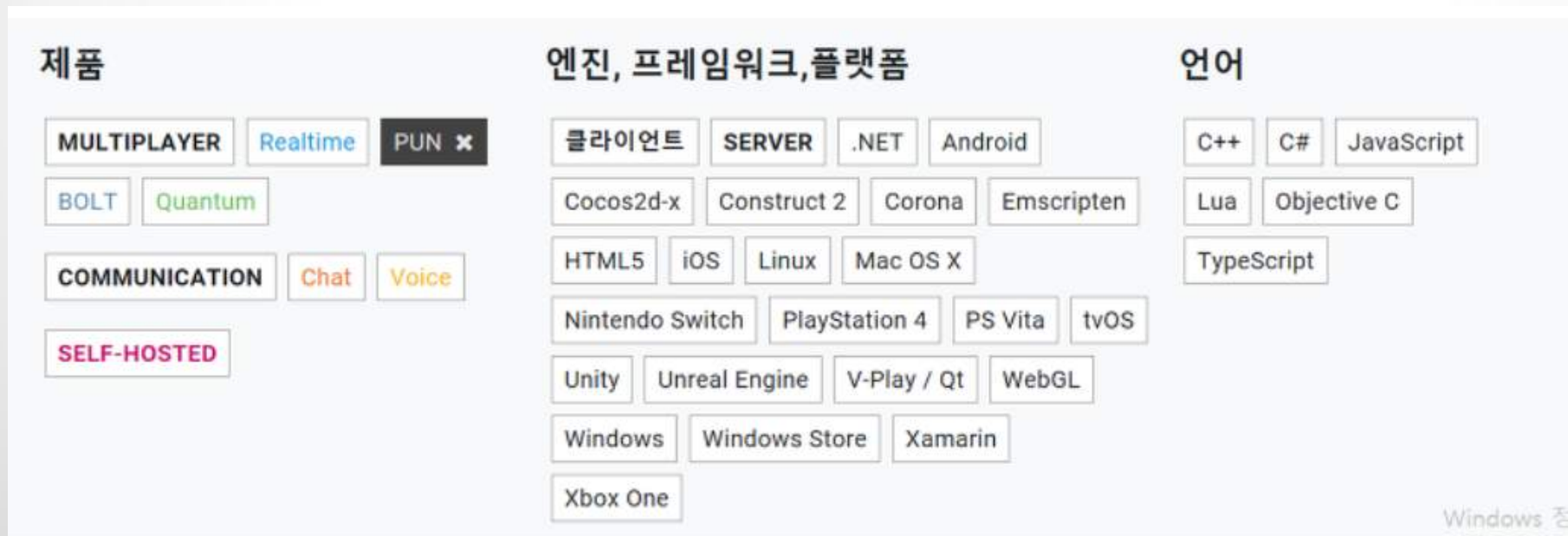
- 유료이지만 아주 강력한 네트워크
- 공부용으로는 무료로 받을수 있음
- <https://www.photonengine.com/ko-KR/Photon>

A screenshot of the Photon Engine website banner. The banner has a dark blue background with a glowing, futuristic circular graphic on the right side. The text is in white and Korean. At the top left is the Photon logo and the word 'photon'. To the right of the logo is a dropdown menu labeled '제품'. Further right are links for 'SDK', '기술문서', and '로그인'. The main headline reads '멀티플레이어를 간단하게 실현합니다!'. Below this is a paragraph: '세계 No.1의 독립 네트워킹 엔진, 멀티 플레이어 플랫폼입니다.- 신속함, 신뢰성, 확장성을 느껴보세요! 인디부터 초대형 게임 스튜디오까지, 모든 분들을 위해 만들어졌습니다.' At the bottom left, it says '935,162 온라인 중인 플레이어'. At the bottom right, there is a black button with white text that says 'PHOTON 무료플랜 사용하기'.

# 포톤네트워크를 사용한 FPS 개발

- 포톤 SDK

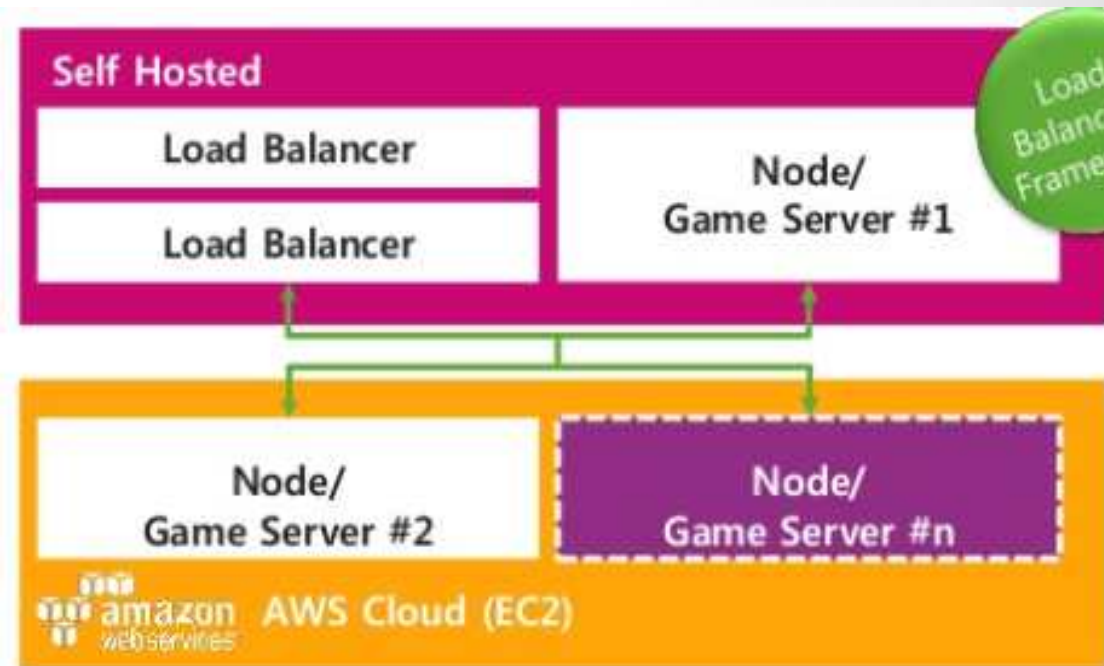
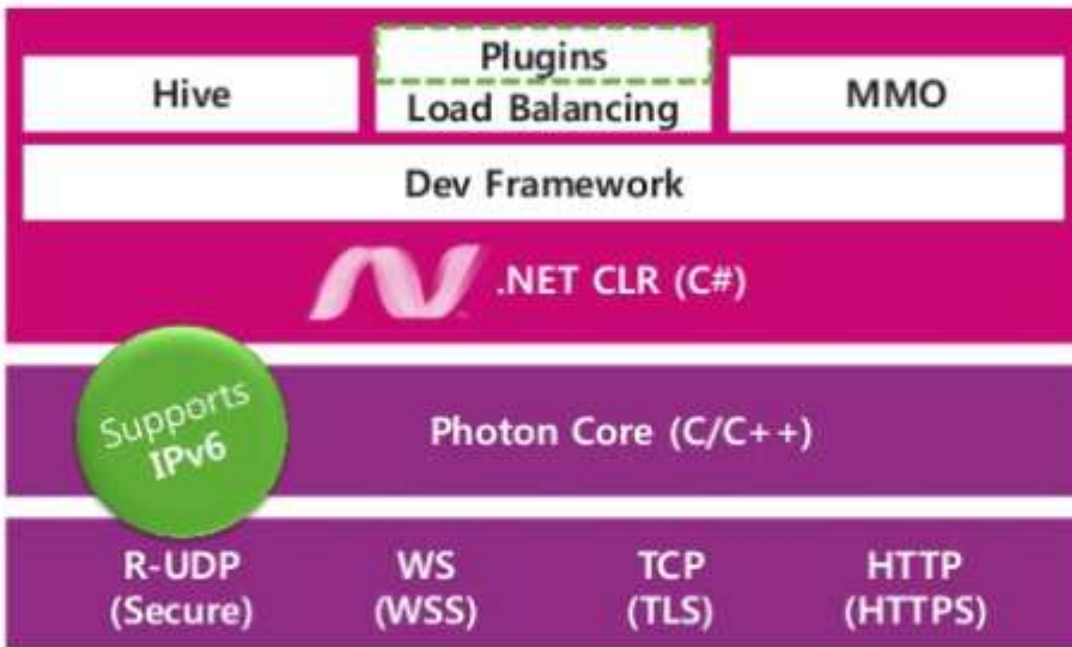
- 다양한 플랫폼에서 멀티 플레이어 게임 개발이 간단히 이루어짐
- UNET이 빠진 2019 버전에서는 현실적인 대안임



# 포톤네트워크를 사용한 FPS 개발

- 포톤서버

- 내부적으로 APP, Web, NAT, VPN 등 네트워크 환경을 고려한 구축을 수행



# 포톤네트워크를 사용한 FPS 개발

- 가입후 절차

- 회원 가입후 관리 화면으로 이동
- 처음에 생성된 애플리케이션 정보는 삭제를 해야 함

이 애플리케이션은 무료 플랜입니다.  
게임 발매 전에 플랜을 업그레이드 해 주십시오.

|          |    |       |
|----------|----|-------|
| 이용 플랜    | 20 | CCU   |
| 최대치 이번 달 | 0  | CCU → |
| 최대치 지난달  | 0  | CCU   |
| 접속 거부 수  | 0  |       |

통계 정보

상세 정보

CCU 변경하기

쿠폰 / PUN+ 추가

# 포톤네트워크를 사용한 FPS 개발

- 가입후 절차

- 새어플리케이션 만들기를 통해서 포톤의 종류와 이름을 입력
- 무료이기 때문에 동시접속 20명까지 가능

## 프로퍼티

이름

\

URL

어플리케이션 설명

Default

동시접속 사용자 수

Subscription

0 CCU

One-time

0 CCU

쿠폰

0 CCU

합계

20 CCU CCU Burst 이용불가

프로퍼티 편집

또는

어플리케이션을 삭제하기

# 포톤네트워크를 사용한 FPS 개발

- 가입후 절차

- 새어플리케이션 만들기를 통해서 포톤의 종류와 이름을 입력
- 무료이기 때문에 동시접속 20명까지 가능

## 사용중인 Photon Cloud 어플리케이션

현재 어플리케이션 셋업이 없습니다.

새 어플리케이션 만들기

# 포톤네트워크를 사용한 FPS 개발

- 가입후 절차

- 새어플리케이션 만들기를 통해서 포톤의 종류와 이름을 입력
- 무료이기 때문에 동시접속 20명까지 가능



The image shows a web form for creating a new Photon application. The form has several fields, some of which are highlighted with red boxes. The fields are:

- Photon 종류 \***: A dropdown menu with "Photon Realtime" selected. This field is highlighted with a red box.
- 이름 \***: A text input field containing "MyFps". This field is highlighted with a red box.
- 어플리케이션 설명**: A text input field containing "수업용 fps 게임".
- URL**: A text input field containing "http://enter.your-url.here/".
- Buttons**: At the bottom, there are two buttons: "작성하기" (Save) and "또는 뒤로가기" (or Go Back). The "작성하기" button is highlighted with a red box.



# 포톤네트워크를 사용한 FPS 개발

- 가입후 절차

- 아이디는 잘 기억하고 복사해 두었다가 유니티 게임을 만들때 사용 해야 합니다

**MyFps**  
어플리케이션 ID: 4a0 ×

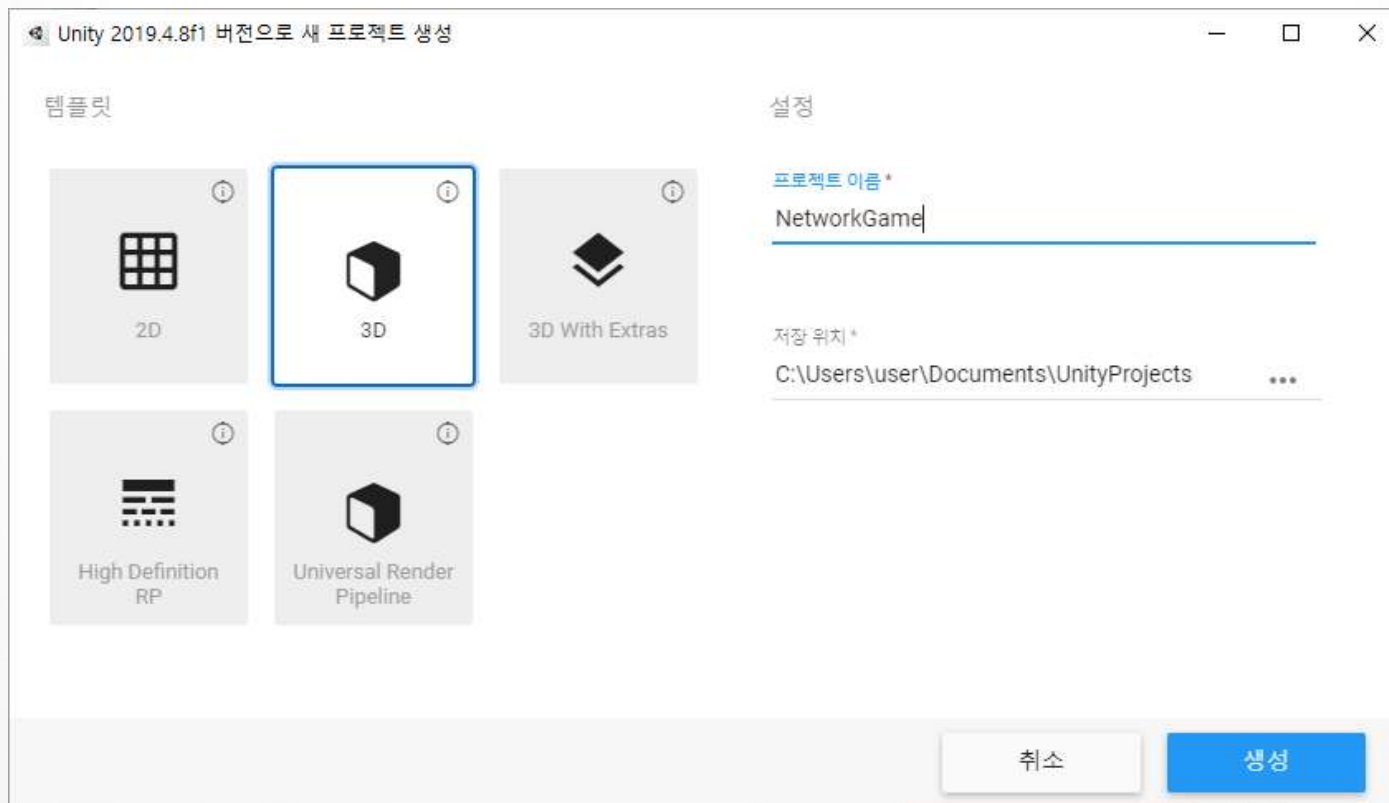
이 어플리케이션은 무료 플랜입니다.  
게임 발매 전에 플랜을 업그레이드 해 주십시오.

|          |    |       |
|----------|----|-------|
| 이용 플랜    | 20 | CCU   |
| 최대치 이번 달 | 0  | CCU → |
| 최대치 지난달  | 0  | CCU   |
| 접속 거부 수  | 0  |       |

통계 정보상세 정보CCU 변경하기쿠폰 / PUN+ 추가

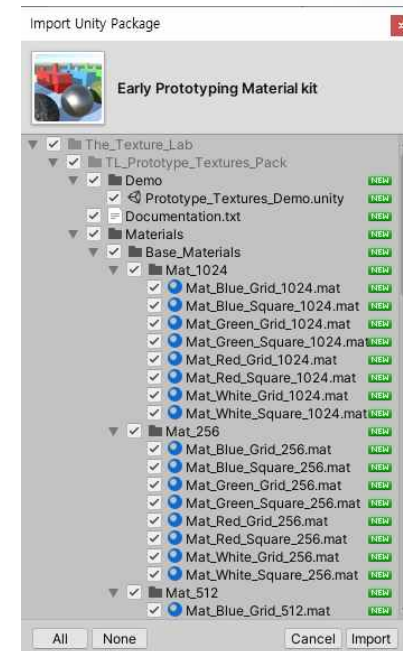
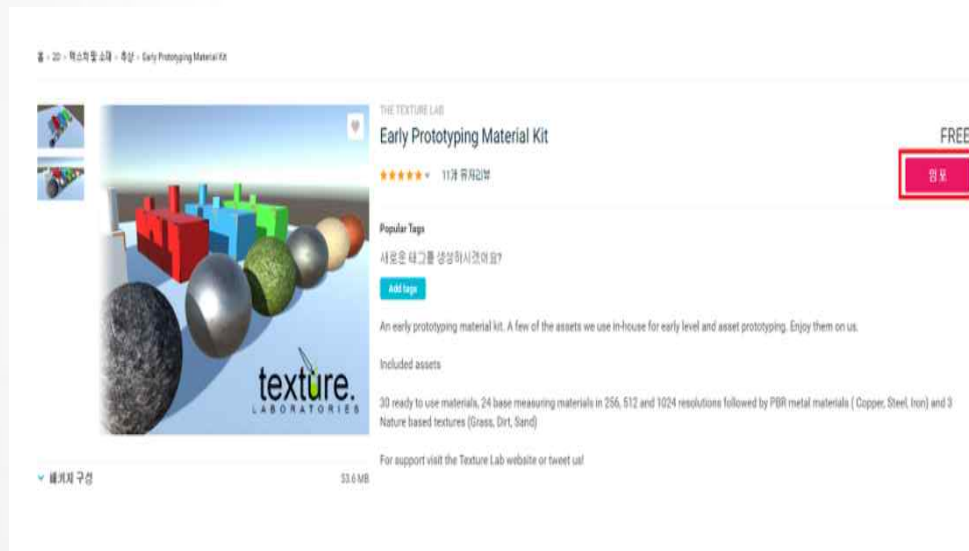
# 포톤네트워크를 사용한 FPS 개발

- NetworkGame 이름으로 프로젝트 생성하기!



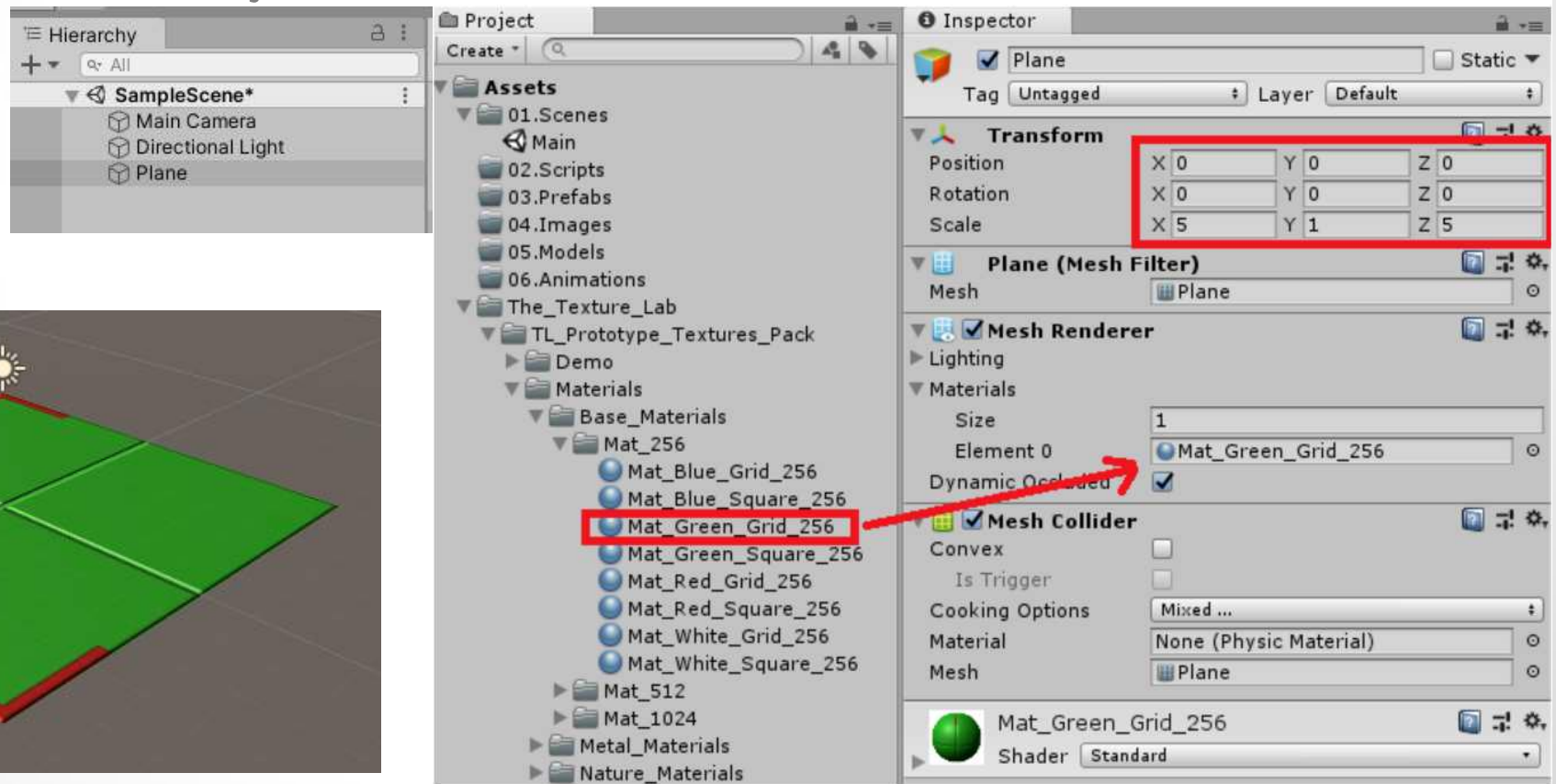
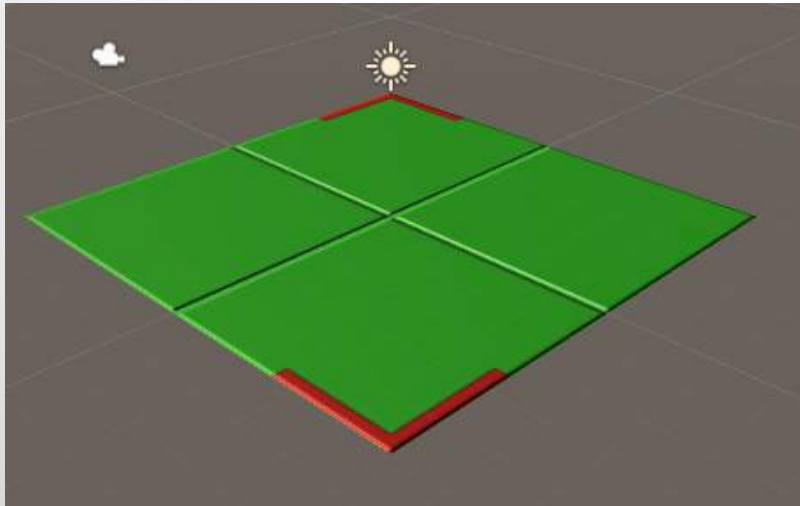
# 포톤네트워크를 사용한 FPS 개발

- Early Prototyping Material Kit 다운로드 및 импорт
  - 바닥 및 캐릭터를 꾸밀수 있음



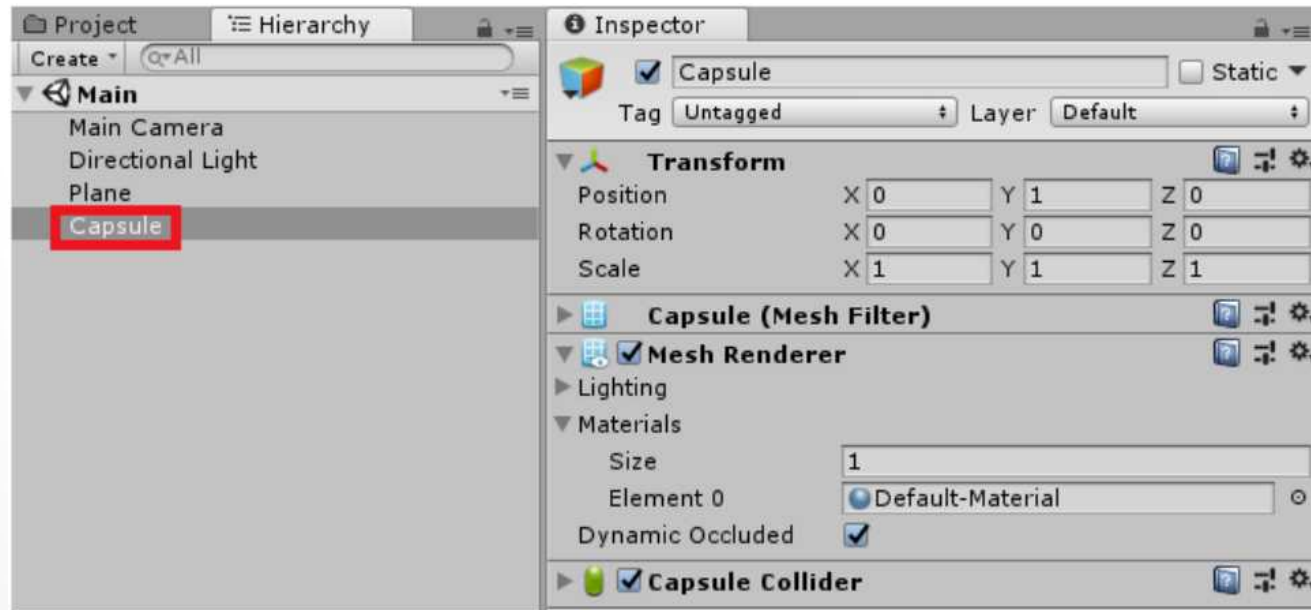
# 포톤네트워드를 사용한 FPS 개발

- 프로젝트에 3D Object -> Plane 을 추가하고 다음과 같이 설정



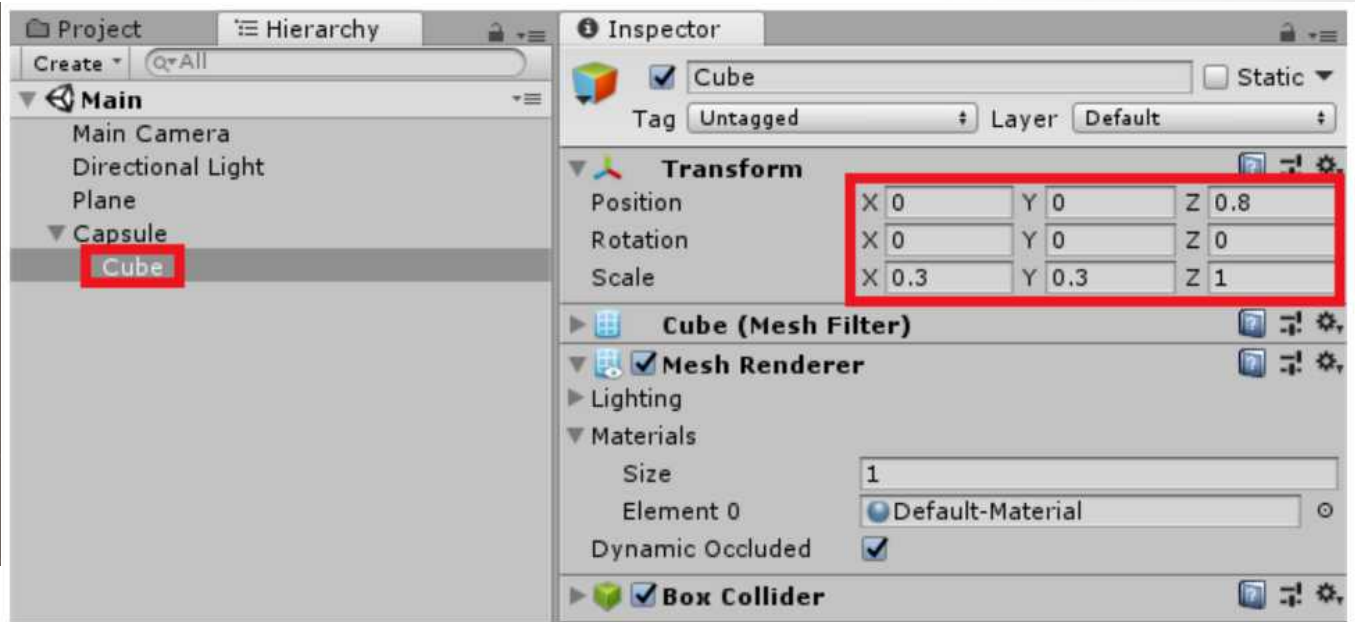
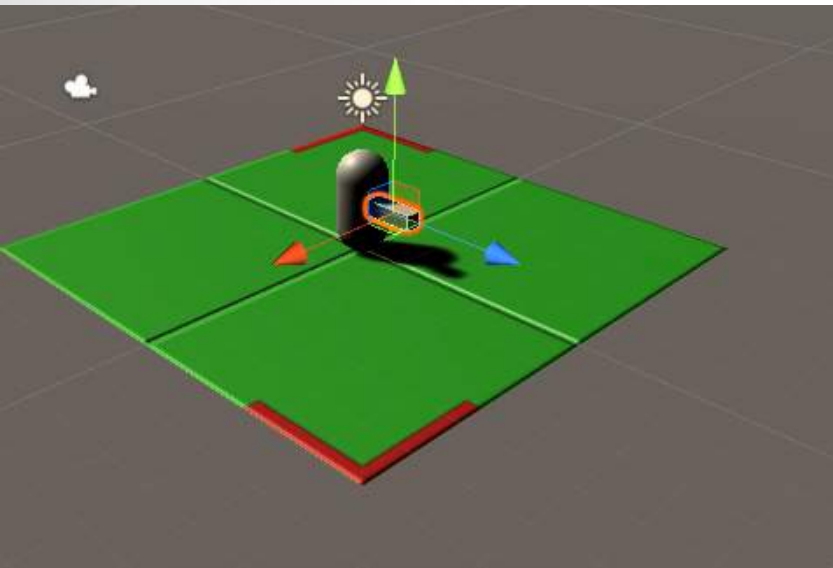
# 포톤네트워크를 사용한 FPS 개발

- 플레이어 만들기
  - 3D오브젝트 -> Capsule 생성후 포지션값 수정
  - 3D오브젝트 -> Cube 생성후 Transform 값을 변경
  - Capsule의 자식으로 Cube를 넣음



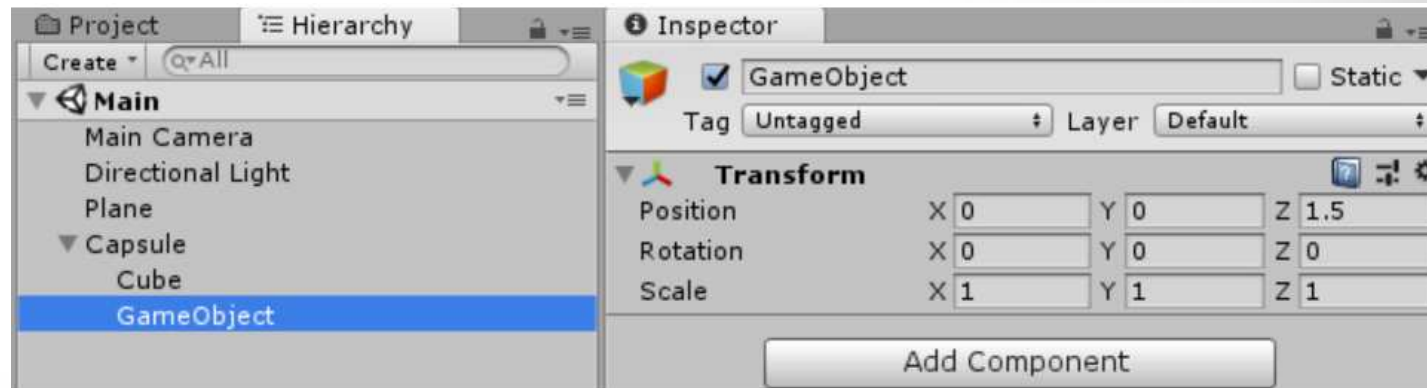
# 포톤네트워크를 사용한 FPS 개발

- 플레이어 만들기
  - 3D오브젝트 -> Capsule 생성후 포지션값 수정
  - 3D오브젝트 -> Cube 생성후 Transform 값을 변경
  - Capsule의 자식으로 Cube를 넣음



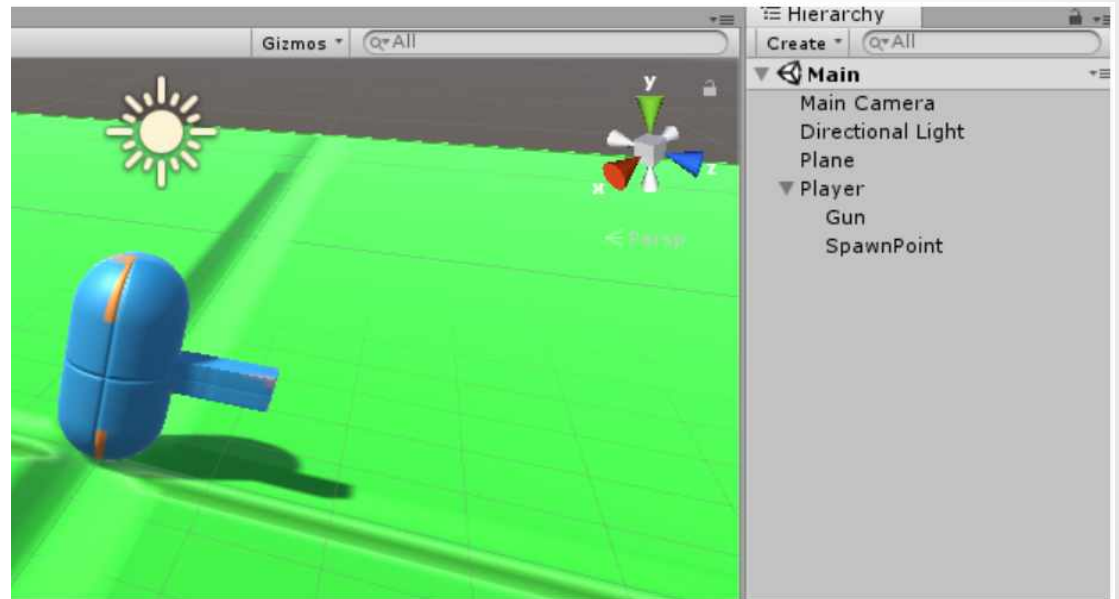
# 포톤네트워크를 사용한 FPS 개발

- 플레이어 만들기
  - 빈 오브젝트 생성후 Position을 다음과 같이 변경
  - Capsule 의 자식으로 넣기
  - 다음과 같이 이름 변경하기
  - Capsule -> Player
  - Cube -> Gun
  - GameObject -> SpawnPoint



# 포톤네트워크를 사용한 FPS 개발

- 플레이어 만들기
  - 빈 오브젝트 생성후 Position을 다음과 같이 변경
  - Capsule 의 자식으로 넣기
  - 다음과 같이 이름 변경하기
  - Capsule -> Player
  - Cube -> Gun
  - GameObject -> SpawnPoint

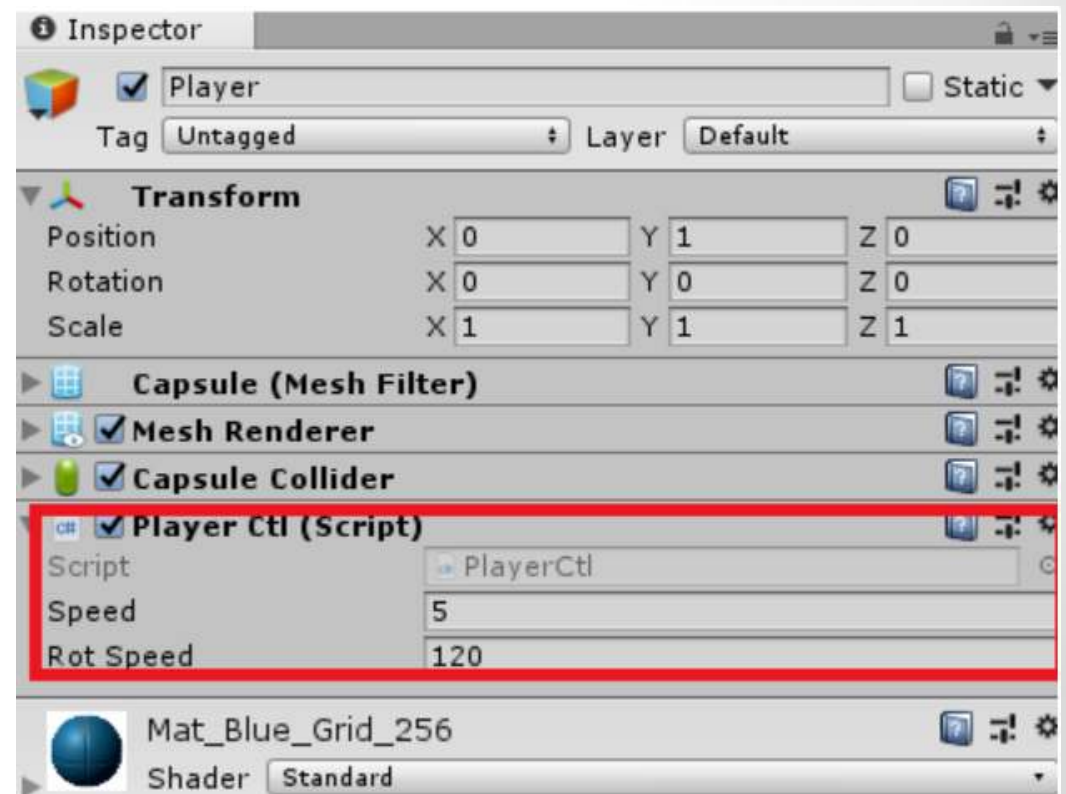




# 포톤네트워크를 사용한 FPS 개발

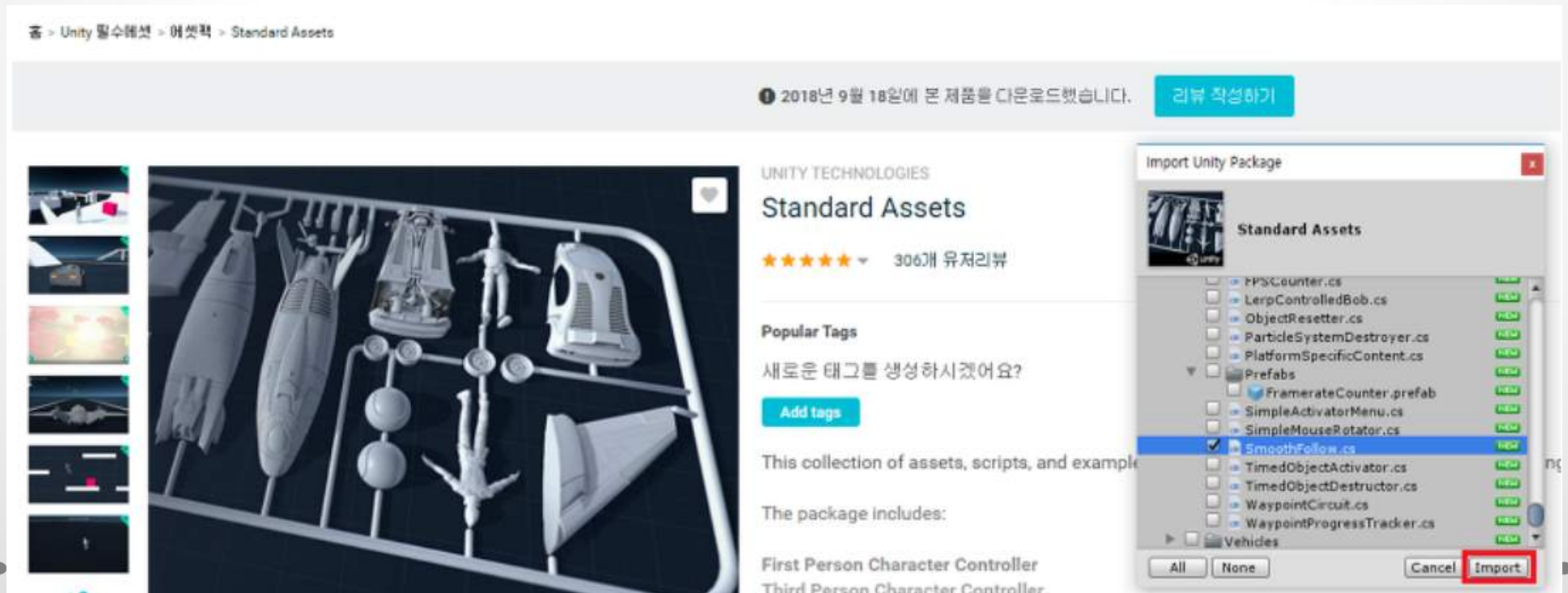
- 플레이어 스크립트 작성
  - 키보드 상하 좌우 값을 입력받고 회전과 이동이 가능

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerCtl : MonoBehaviour {
6
7     public float speed = 5.0f;
8     public float rotSpeed = 120.0f;
9
10    private Transform tr;
11
12    void Start()
13    {
14        tr = GetComponent<Transform>();
15    }
16
17    void Update()
18    {
19        float h = Input.GetAxis("Horizontal");
20        float v = Input.GetAxis("Vertical");
21
22        tr.Translate(Vector3.forward * v * Time.deltaTime * speed);
23        tr.Rotate(Vector3.up * h * Time.deltaTime * rotSpeed);
24    }
25 }
```

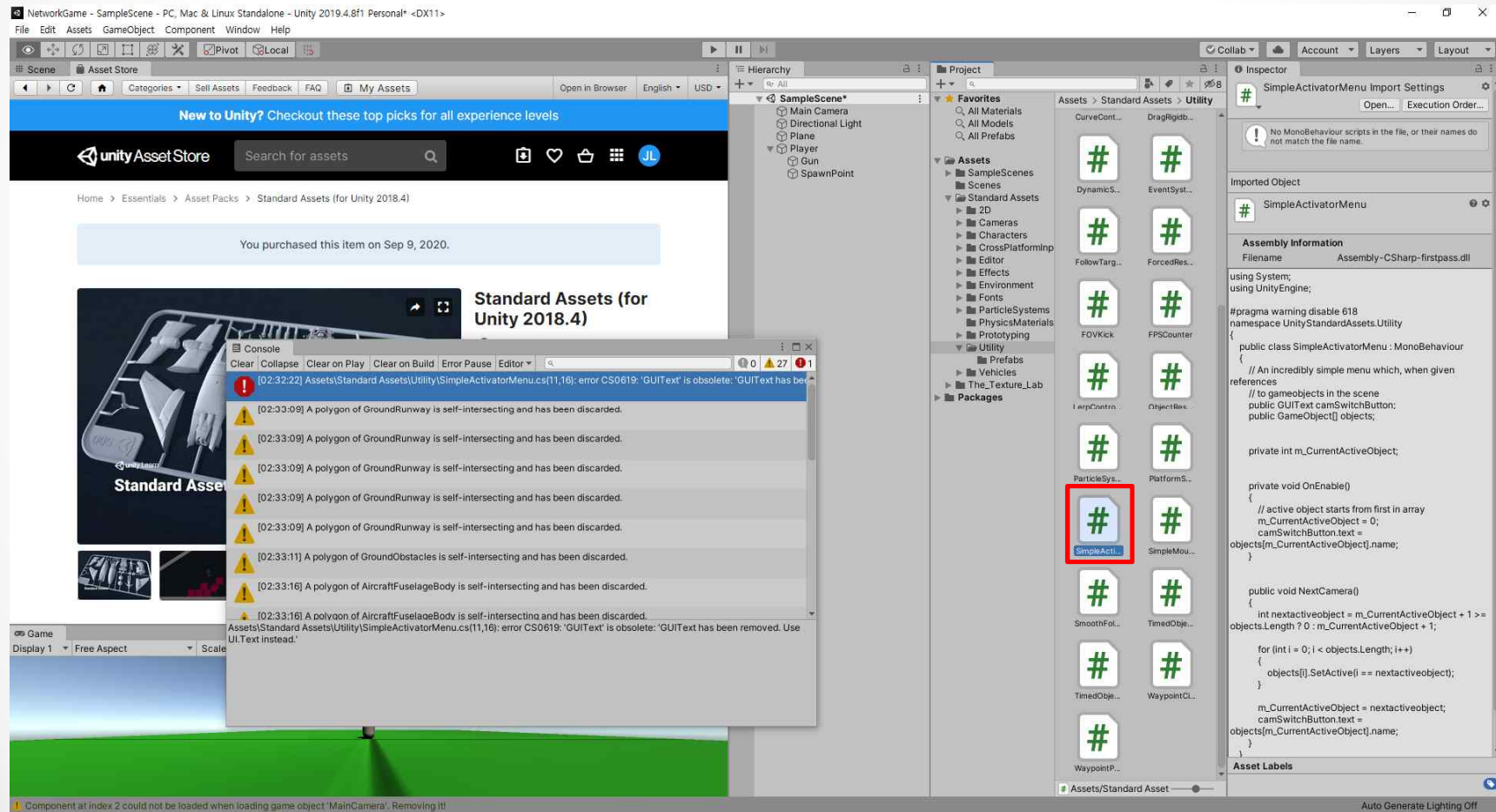


# 포톤네트워크를 사용한 FPS 개발

- SmoothFollow 스크립트
  - 카메라가 플레이어를 자연스럽게 따라갈 수 있음
  - 애셋스토어에서 StandardAssets를 임포트하여 SmoothFollow 스크립트를 받기

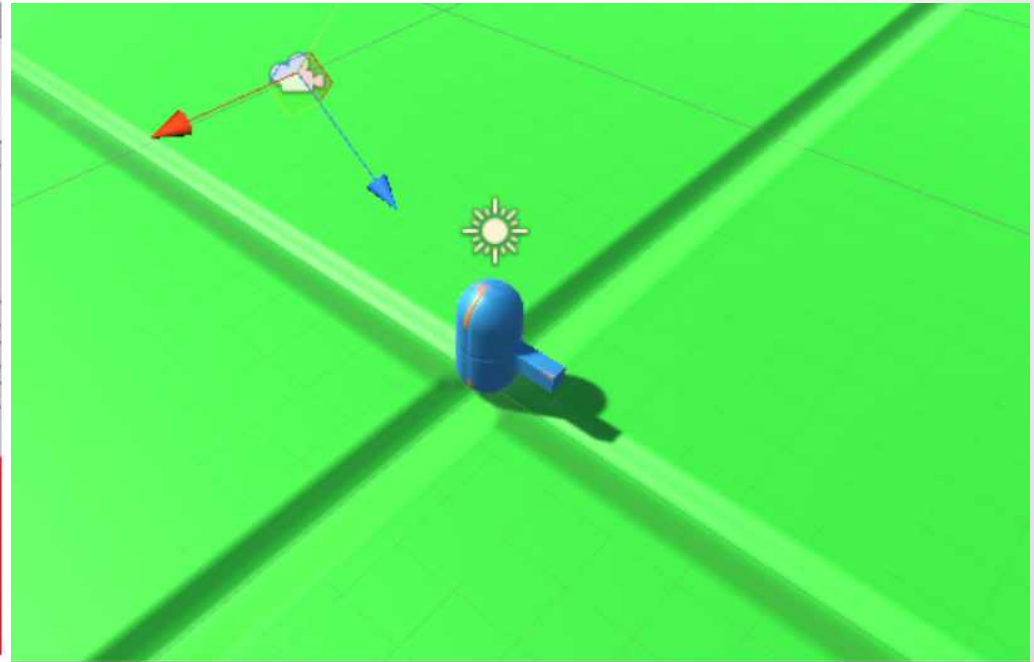
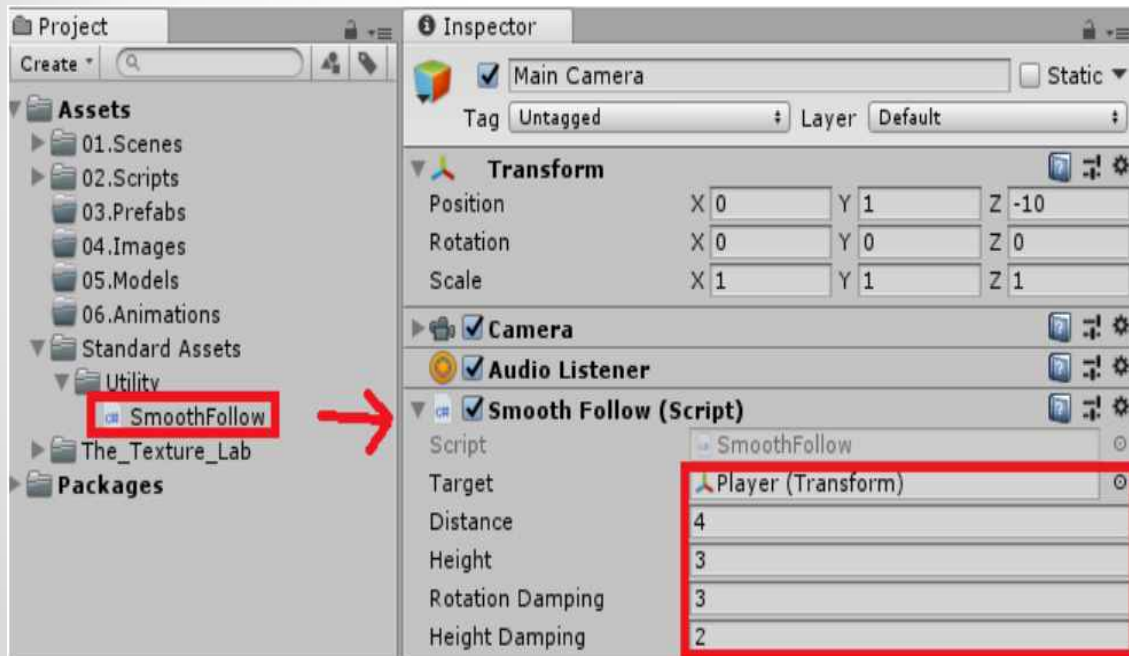


# 포톤네트워크를 사용한 FPS 개발



# 포톤네트워크를 사용한 FPS 개발

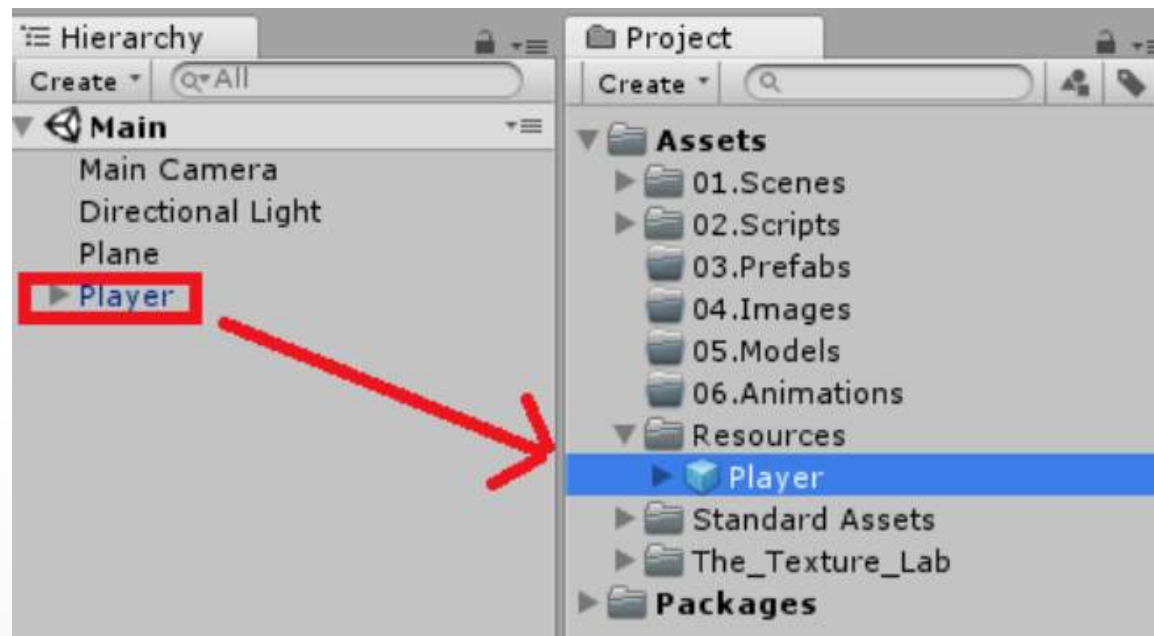
- SmoothFollow 스크립트
  - 스크립트를 카메라에 적용하고 컴포넌트 값을 변경



# 포톤네트워크를 사용한 FPS 개발

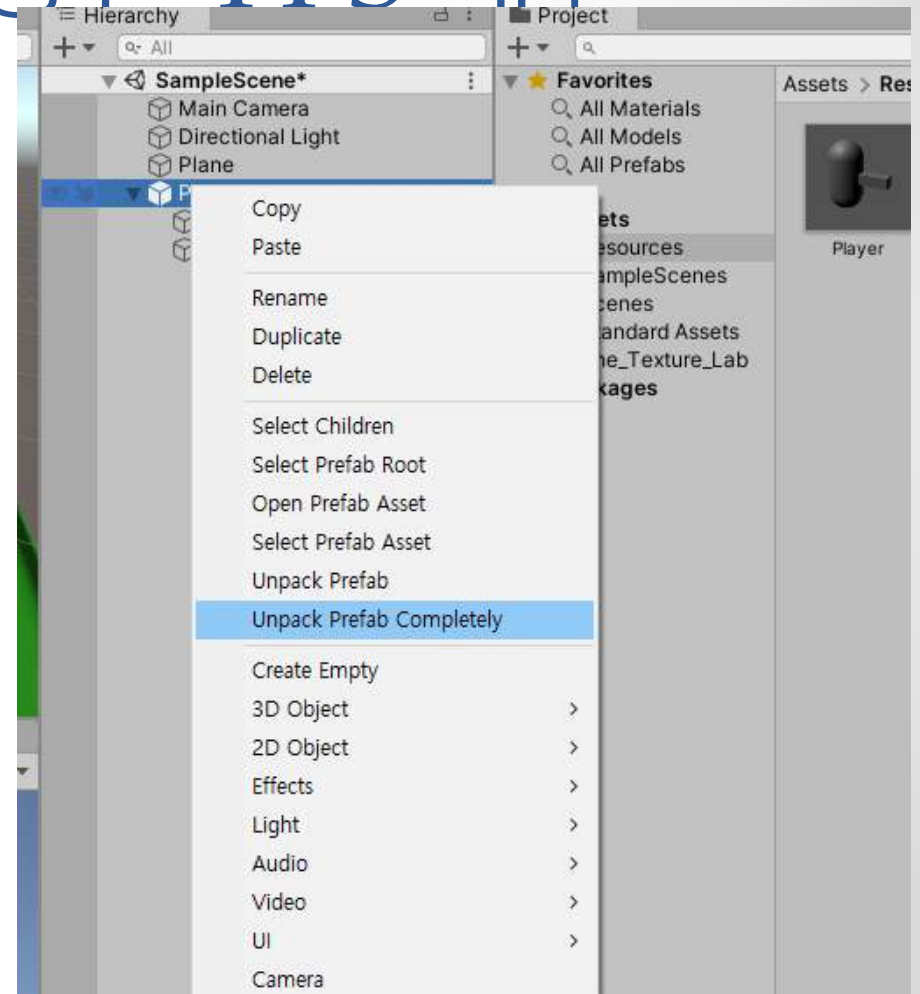
- 프리팸 만들기

- 완성된 플레이어는 Resources 폴더를 만들고 프리팸으로 만들기
- 포톤에서 이 Resources 폴더를 사용함



# 포톤네트워크를 사용한 FPS 개발

- 적 캐릭터 만들기
  - 플레이어 프리팹을 Unpack Prefab Completely 기능을 이용해서 프리팹을 해제
  - Player -> Enemy로 변경

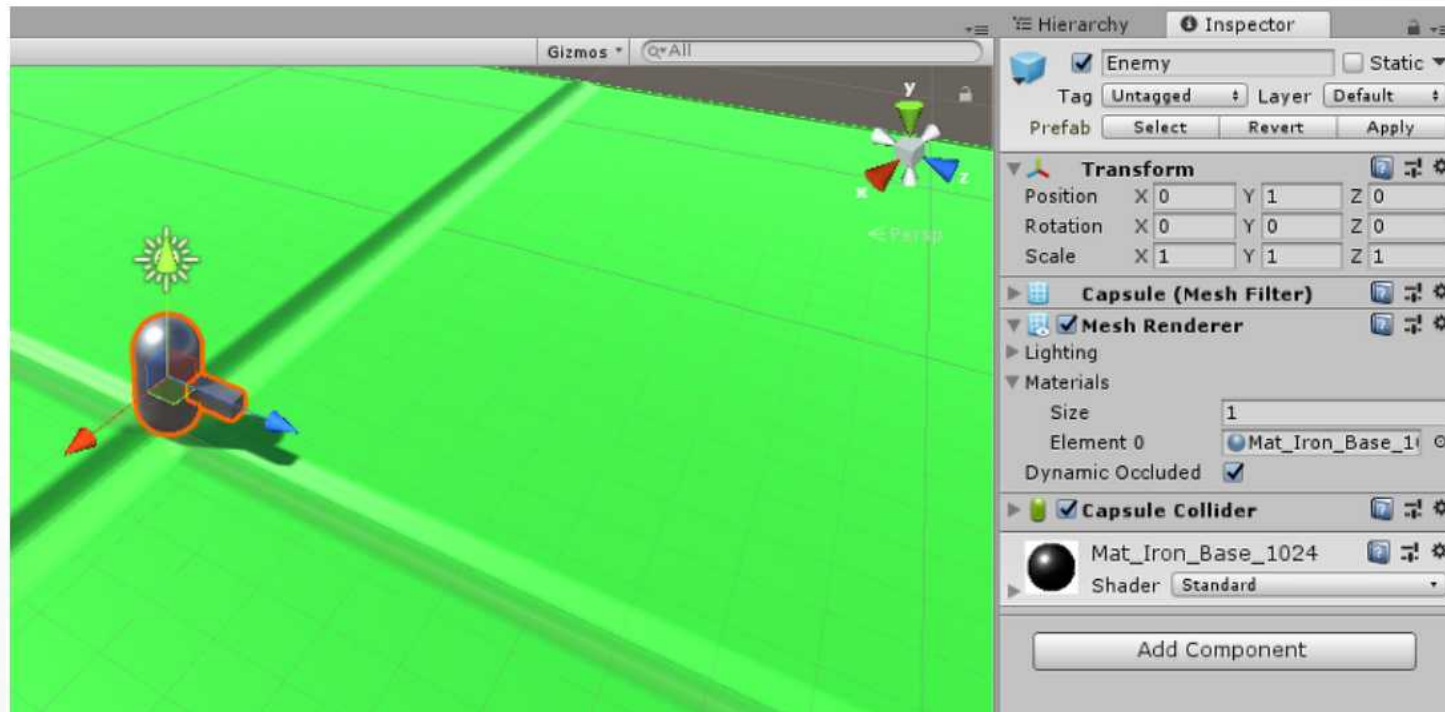




# 포톤네트워크를 사용한 FPS 개발

- 적 캐릭터 만들기

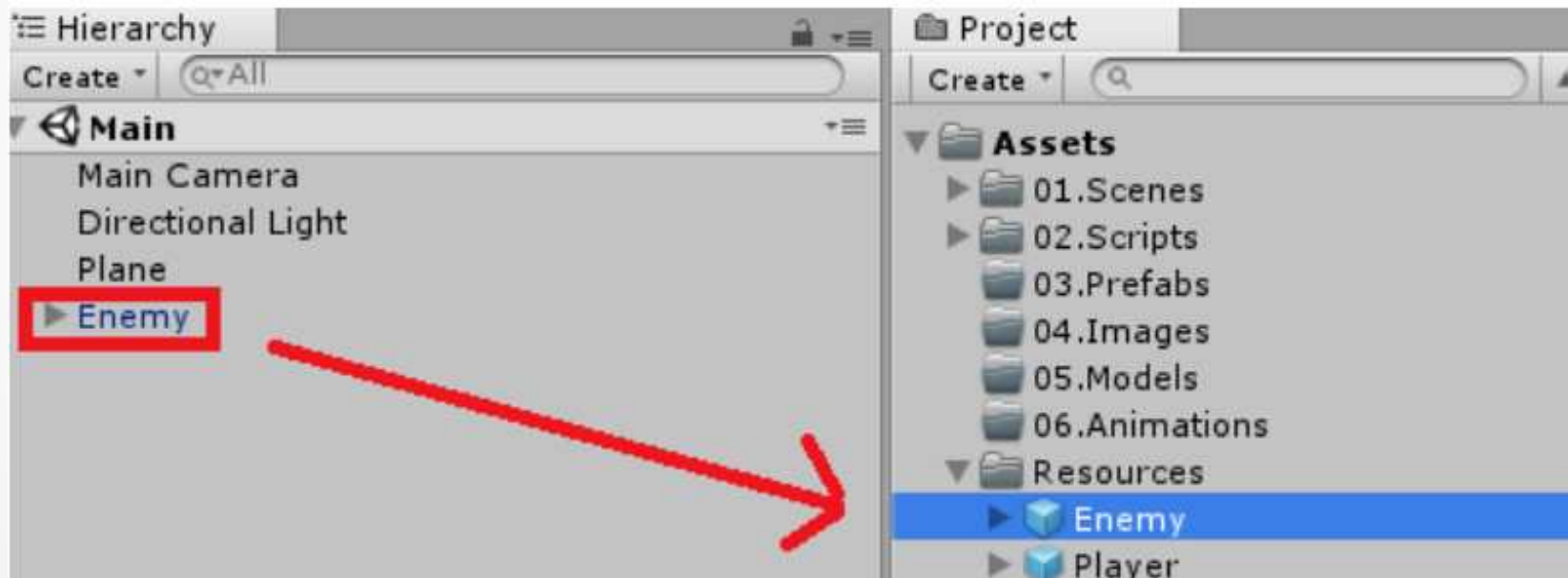
- Enemy 객체에 있는 Player 스크립트를 없애고, 머티리얼을 다음과 같이 변경해보기 (플레이어와 구분이 필요함)



# 포톤네트워크를 사용한 FPS 개발

- 적 캐릭터 만들기

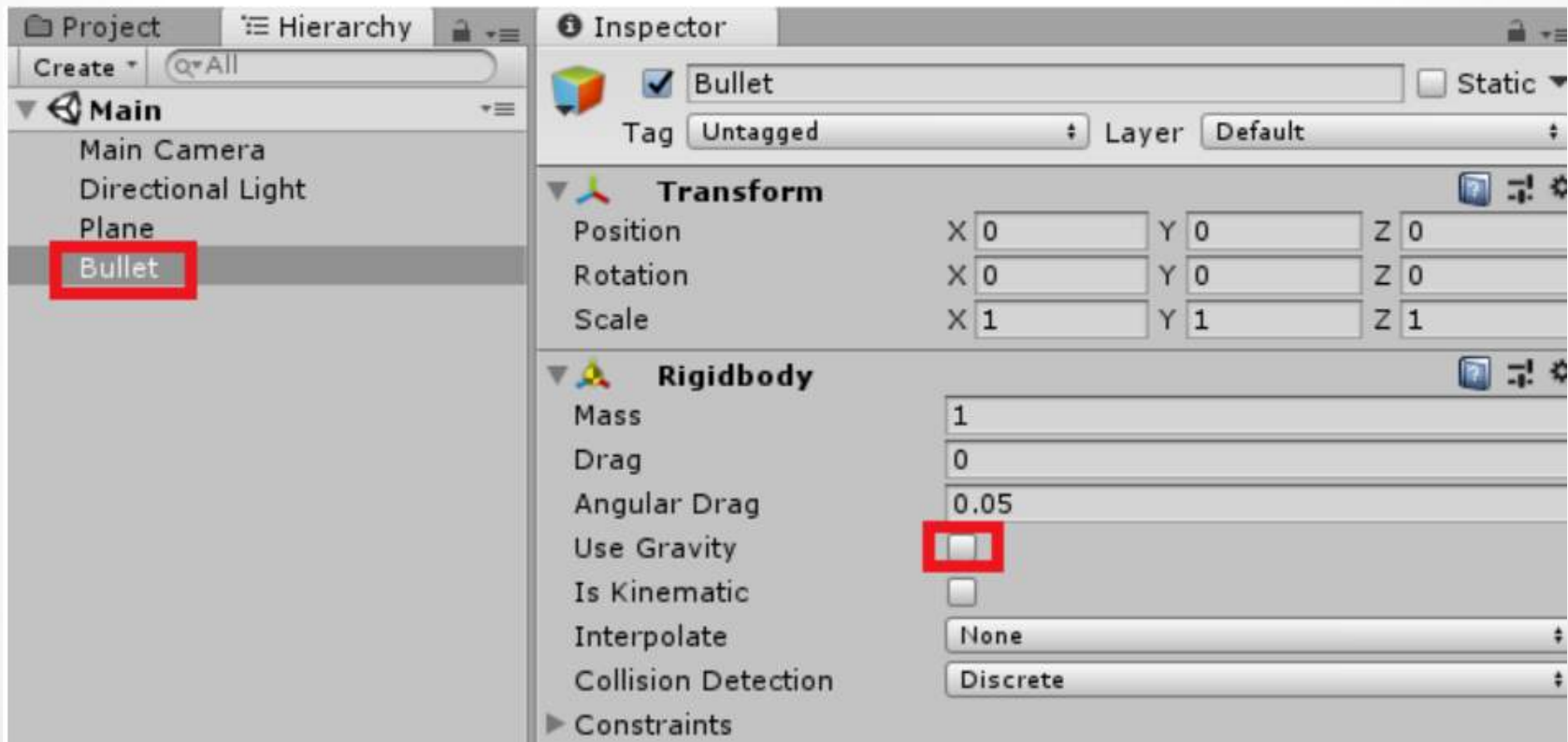
- 완성된 Enemy 객체를 Resources 폴더에 넣어 프리팹으로 만들어야 함





# 포톤네트워크를 사용한 FPS 개발

- 총알 만들기
  - 총알을 만들때 Rigidbody를 추가하고, 중력 체크는 해제



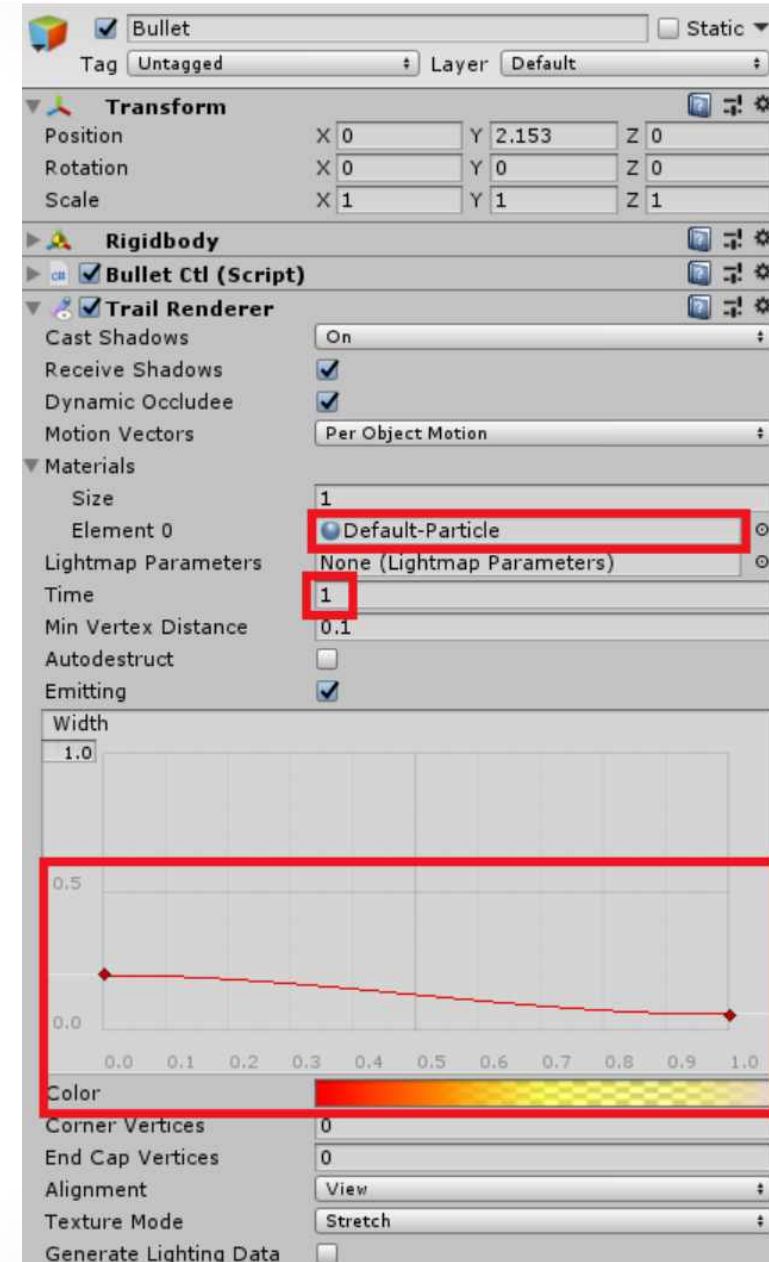
# 포톤네트워크를

- BulletCtrl.cs 파일 생성
  - spawnPoint를 기준으로 총알이 만들어지고, 일정 거리가 지나가면 없어지도록 되어 있음

```
1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4
5  public class BulletCtrl : MonoBehaviour {
6
7      public float speed = 10.0f;
8      public float fireRange = 300.0f;
9      public float damage = 10.0f;
10
11     private Transform tr;
12     private Vector3 spawnPoint;
13
14     void Start()
15     {
16         tr = this.GetComponent<Transform>();
17         spawnPoint = tr.position;
18     }
19
20     void Update()
21     {
22         tr.Translate(Vector3.forward * Time.deltaTime * speed);
23
24         if ( (spawnPoint - tr.position).sqrMagnitude > fireRange)
25         {
26             StartCoroutine(this.DestroyBullet());
27         }
28     }
29
30     IEnumerator DestroyBullet()
31     {
32         Destroy(this.gameObject);
33         yield return null;
34     }
35 }
```

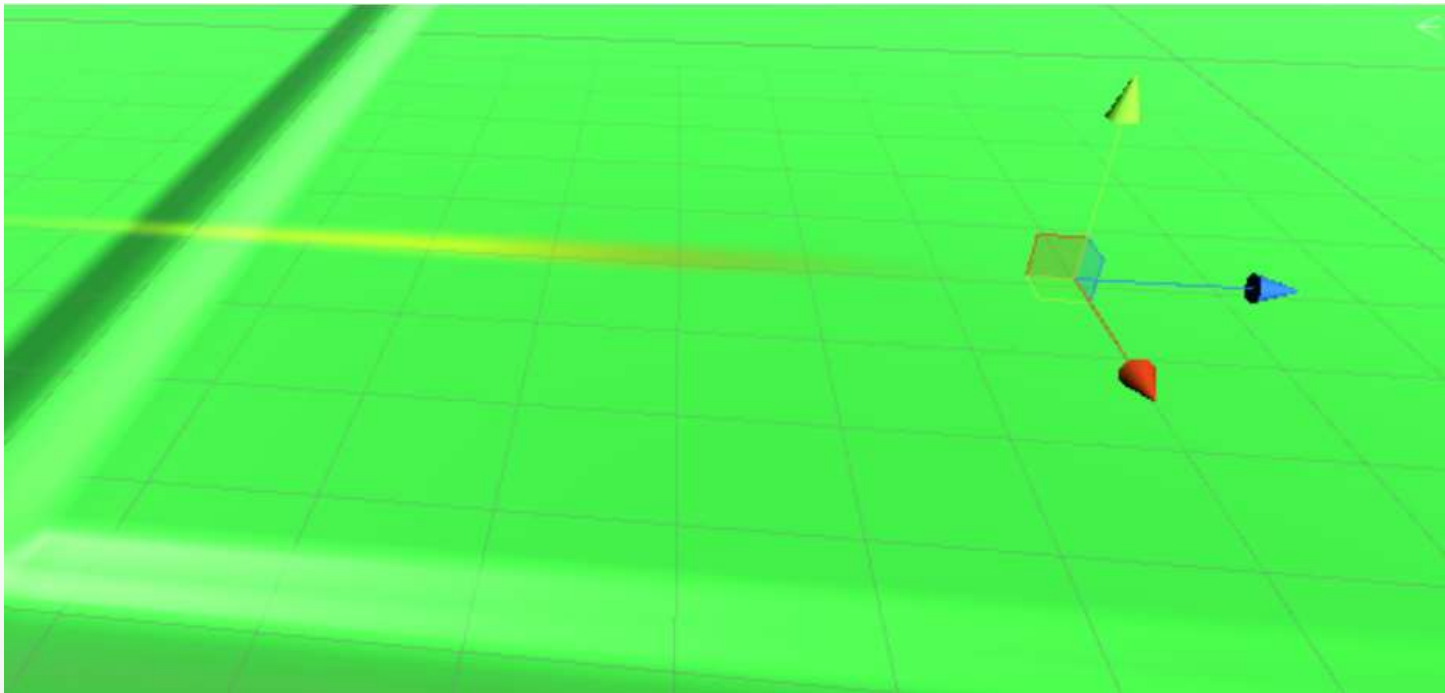
# 포톤네트워크를 사용한

- 총알 효과 적용
  - 총알의 방향을 보여주기 위해 Trail Renderer 추가
  - 머터리얼 수정하기



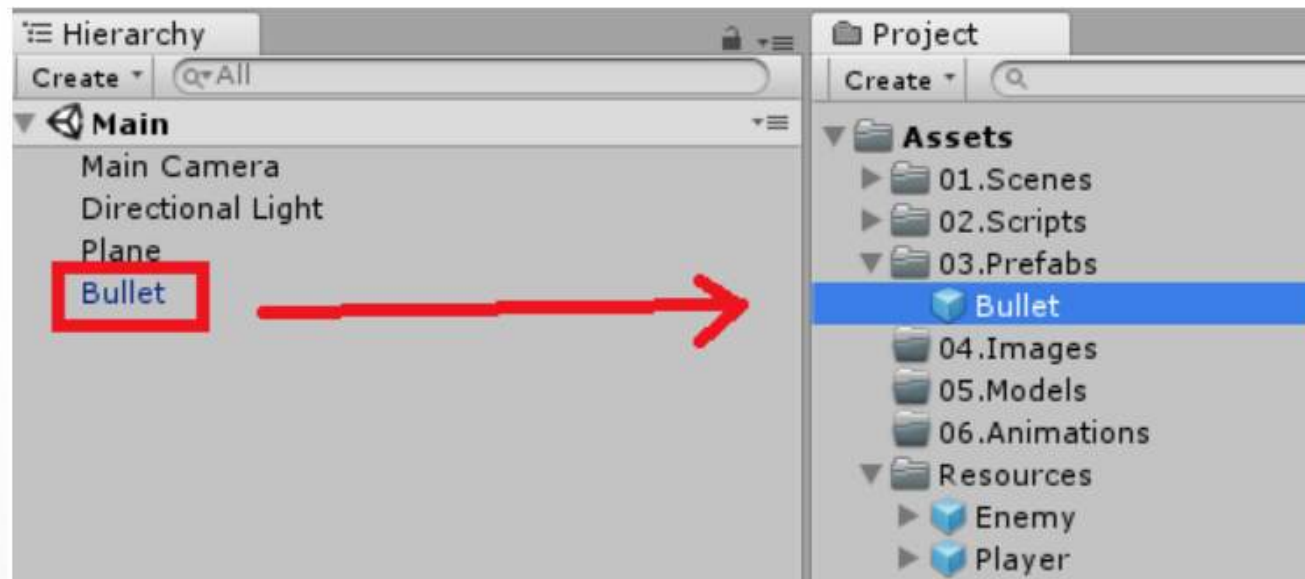
# 포톤네트워크를 사용한 FPS 개발

- 총알 효과 적용
  - 게임 실행후 총알이 날아가는 것을 확인 가능합니다



# 포톤네트워크를 사용한 FPS 개발

- 총알 프리팹 설정
  - 총알은 Prefabs 폴더로 옮겨주기



# 포톤네트워크를 사용한 FPS 개발

- 포톤네트워크 설치
  - 애셋스토어에서 다운 받아서 임포트하기 – 무료버전 받기!



## Photon Unity Networking Classic - FREE

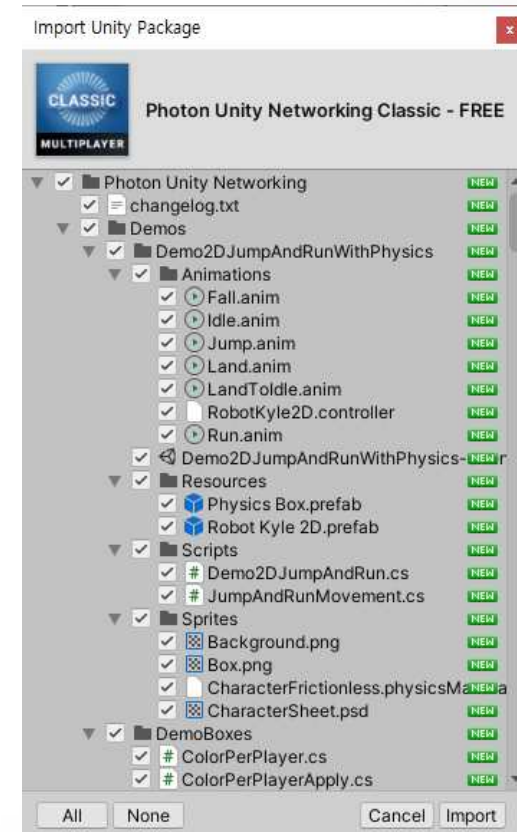
Exit Games  
★★★★★ 5 | 171 Reviews

FREE

Download



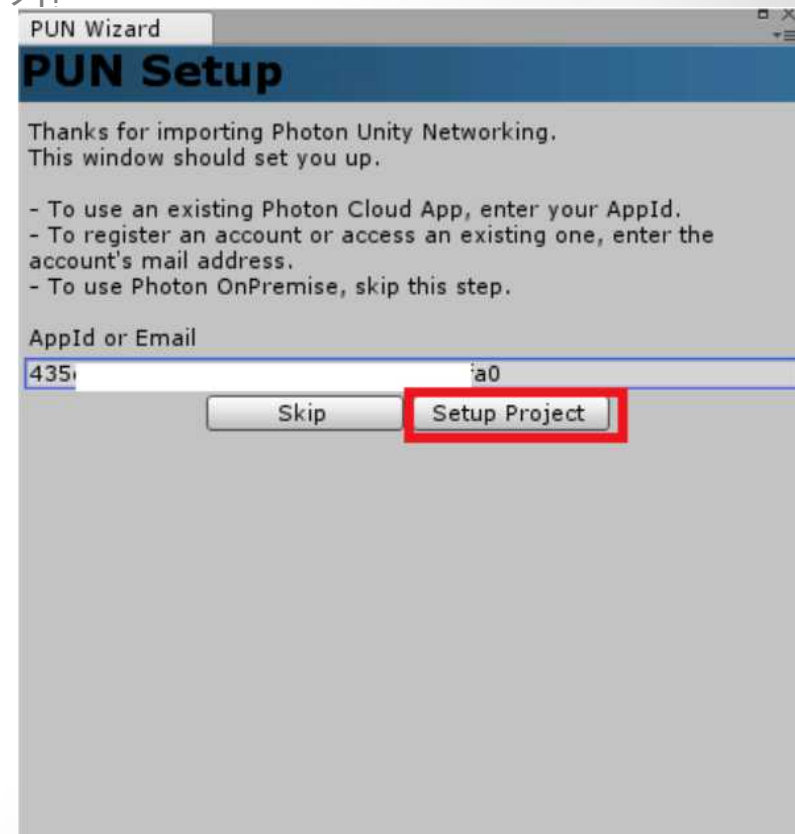
License [Extension Asset](#)  
File size 30.3 MB  
Latest version 1.103.1  
Latest release date Apr 30, 2020  
Support Unity versions 2017.4.7 or higher  
Support [Visit site](#)



# 포톤네트워킹을 사용한 FPS 개발

- 포톤네트워킹 설치

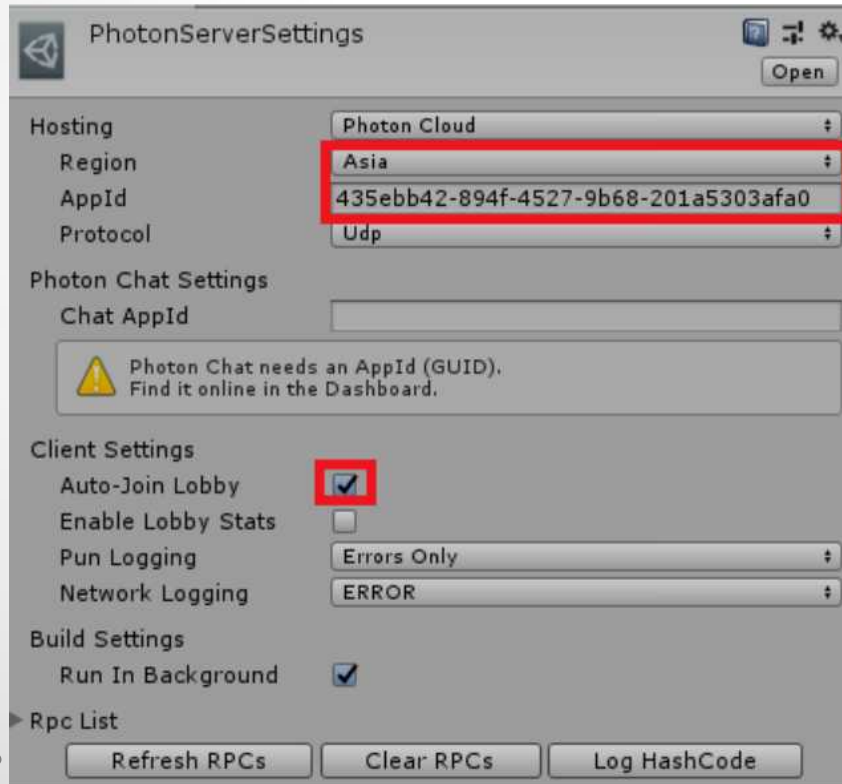
- 설치후 다음 문구를확인하고, 홈페이지에서 가입했던 ID 적용하기!





# 포톤네트워크를 사용한 FPS 개발

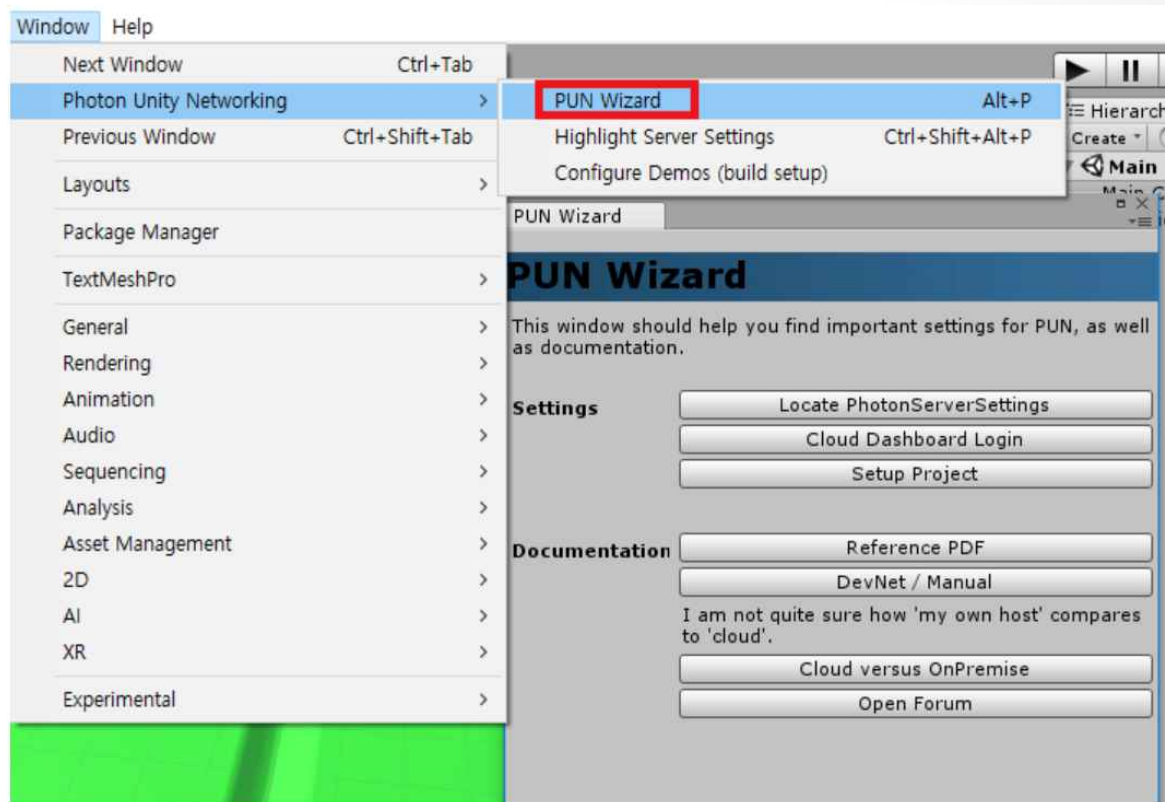
- 포톤네트워크 설정
  - 지역을 Asia로 설정하고, 자동으로 방에 들어 가지도록 만들기





# 포톤네트워킹을 사용한 FPS 개발

- 포톤네트워킹 설정
  - 편 위자드 메뉴에서 추가적인 설정이 가능



# 포톤네트워크를 사용한 FPS 개발

- 방 만들기

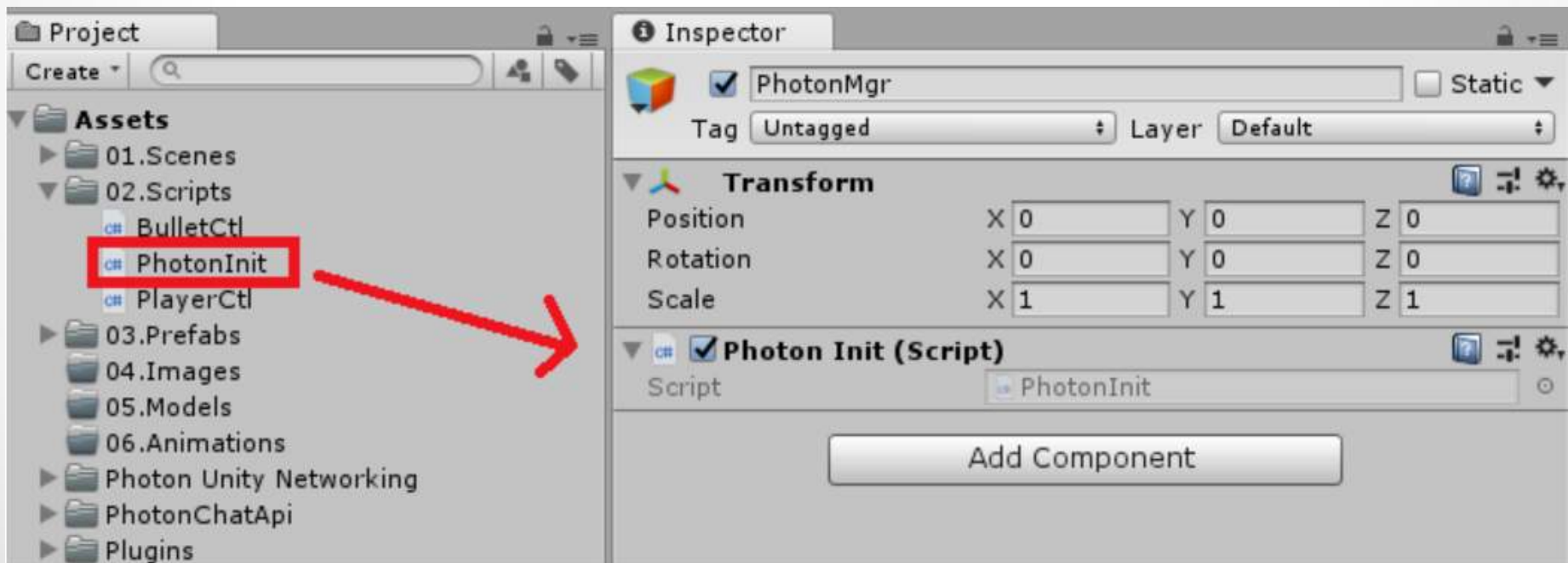
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon;
5
6 public class PhotonInit : Photon.PunBehaviour {
7
8     void Awake()
9     {
10         // 포톤네트워크에 버전별로 분리하여 접속한다.
11         PhotonNetwork.ConnectUsingSettings("MyFps 1.0");
12     }
13
14     // 로비에 입장하였을 때 호출되는 콜백함수
15     public override void OnJoinedLobby()
16     {
17         Debug.Log("Joined Lobby");
18         PhotonNetwork.JoinRandomRoom();
19     }
20
21     // 랜덤 룸 입장에 실패하였을 때 호출되는 콜백함수
22     public override void OnPhotonRandomJoinFailed(object[] codeAndMsg)
23     {
24         Debug.Log("No Room");
25         PhotonNetwork.CreateRoom("MyRoom");
26     }
27 }
```

```
28 // 룸을 생성완료 하였을 때 호출되는 콜백함수
29 public override void OnCreatedRoom()
30 {
31     Debug.Log("Finish make a room");
32 }
33
34 // 룸에 입장되었을 경우 호출되는 콜백함수
35 public override void OnJoinedRoom()
36 {
37     Debug.Log("Joined room");
38 }
39
40 void OnGUI()
41 {
42     GUILayout.Label(PhotonNetwork.connectionStateDetailed.ToString());
43 }
44 }
```

# 포톤네트워크를 사용한 FPS 개발

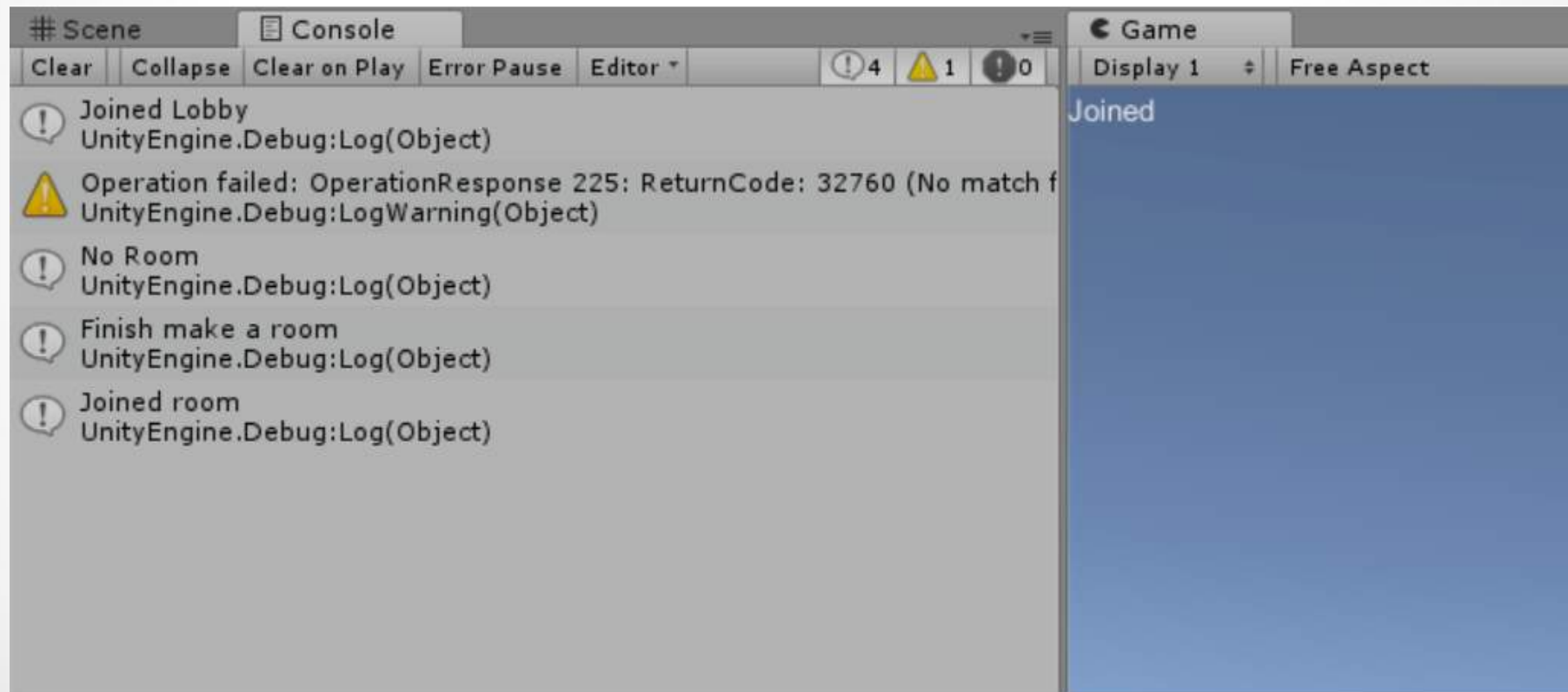
- 방 만들기

- 빈 게임 오브젝트 PhotonMgr로 설정하고 Photon Init 스크립트를 넣어주기
- 포톤네트워크를 활성화하고 룸을 생성할 때 여러가지 콜백 함수가 일어나는걸 확인할 수 있음



# 포톤네트워크를 사용한 FPS 개발

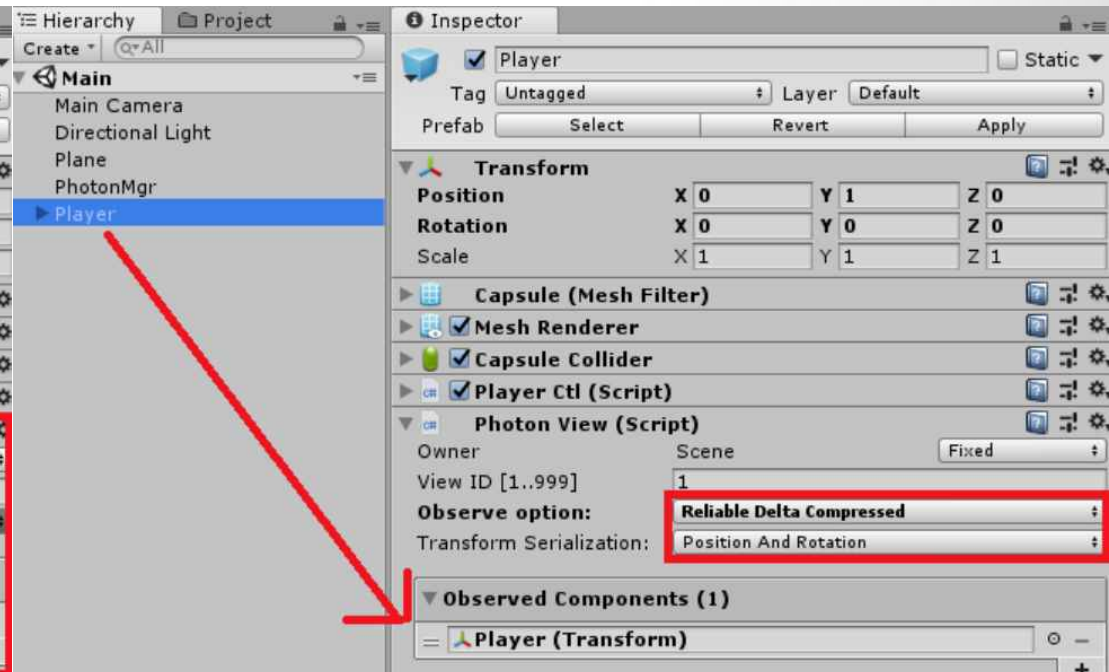
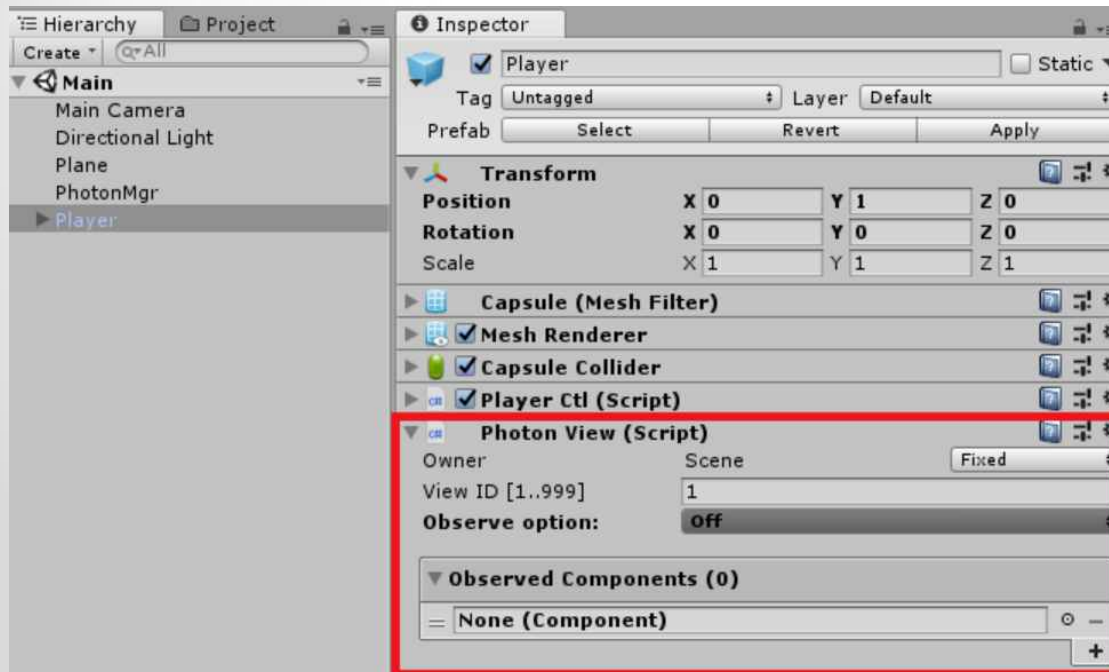
- 방 만들기



# 포톤네트워크를 사용한 FPS 개발

## • 포톤 뷰

- Photon View 스크립트 추가하기
- 네트워크 안에 있는 유저끼리 정보를 공유해줌
- 플레이어 프리팹을 넣어주면 상대방도 내가 보임



# 포톤네트워크를 사용한 FPS 개발

- 포톤 뷰

- Photon View 스크립트 추가하기
- 네트워크 안에 있는 유저끼리 정보를 공유해줌
- 플레이어 프리팹을 넣어주면 상대방도 내가 보임
- 다시 플레이어 프리팹에 설정을 해주자!

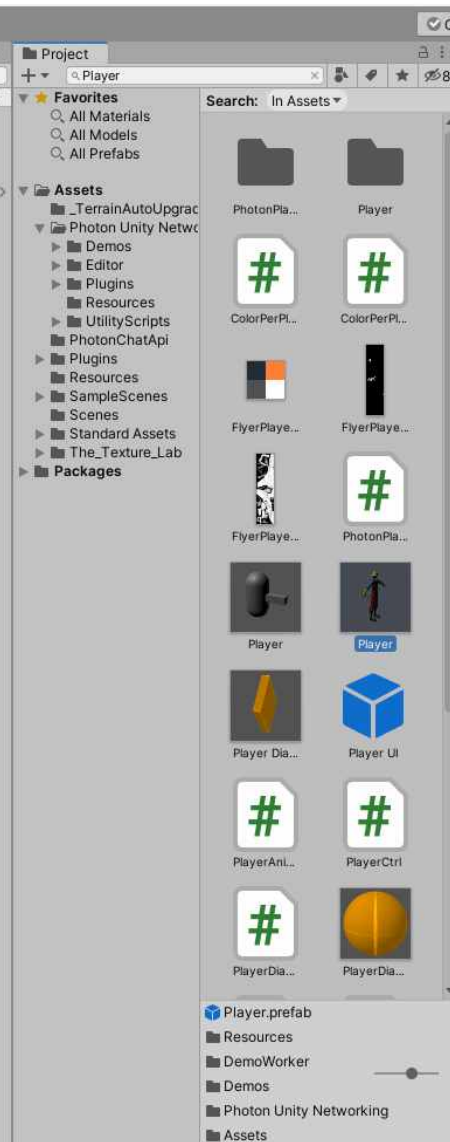
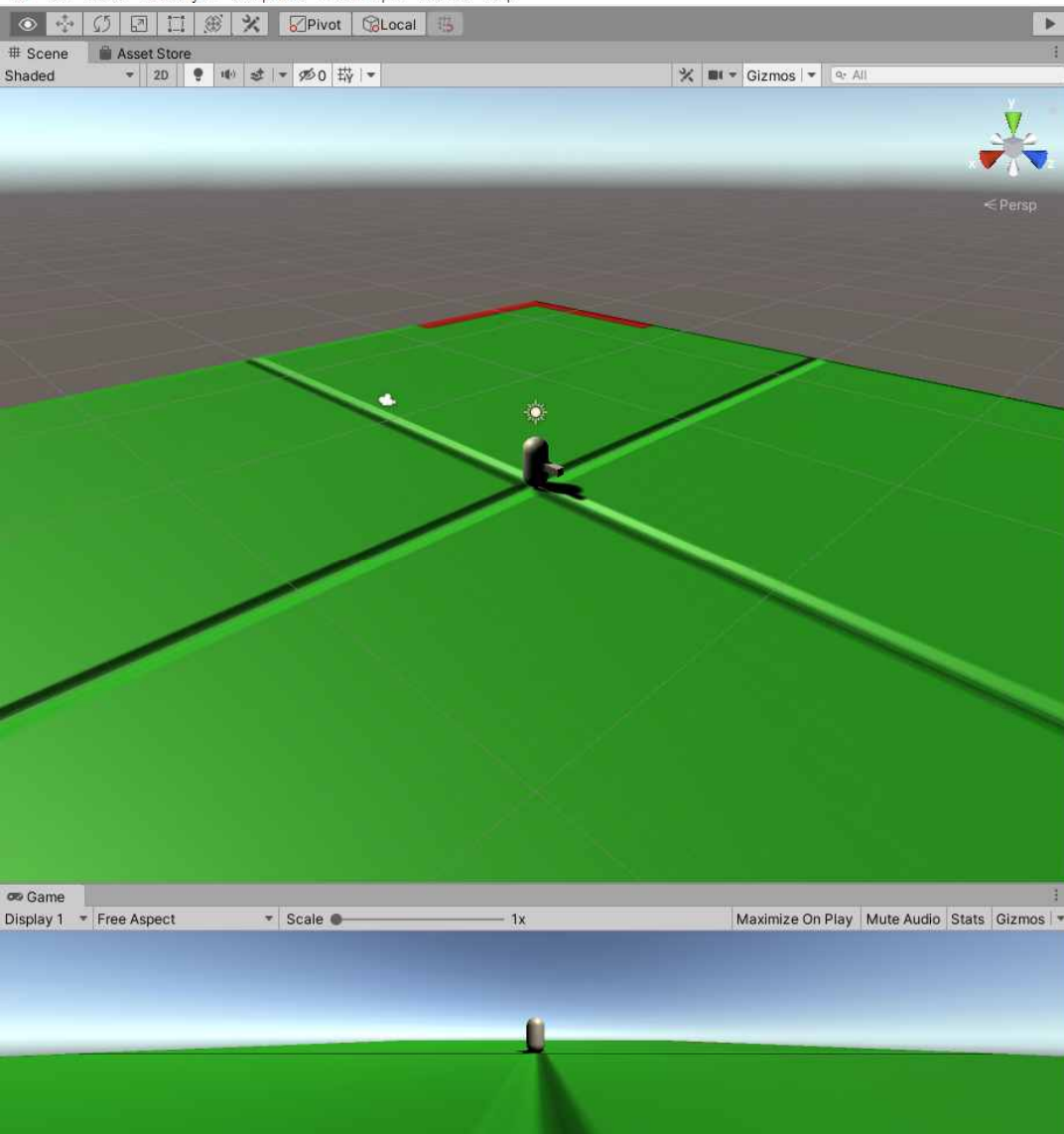


# 포톤네트워크를

- PhotonInit 스크립트 수정
  - 룸에 입장후 정해진 위치에 캐릭터 생성

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon;
5
6 public class PhotonInit : Photon.PunBehaviour {
7
8     void Awake()...
9
10    // 로비에 입장하였을 때 호출되는 콜백함수
11    public override void OnJoinedLobby()...
12
13    // 랜덤 룸 입장에 실패하였을 때 호출되는 콜백함수
14    public override void OnPhotonRandomJoinFailed(object[] codeAndMsg)...
15
16    // 룸을 생성완료 하였을 때 호출되는 콜백함수
17    public override void OnCreatedRoom()...
18
19    // 룸에 입장되었을 경우 호출되는 콜백함수
20    public override void OnJoinedRoom()
21    {
22        Debug.Log("Joined room");
23
24        // 플레이어를 생성한다.
25        StartCoroutine(this.CreatePlayer());
26    }
27
28    // 네트워크상에 연결되어 있는 모든 클라이언트에 플레이어를 생성한다.
29    IEnumerator CreatePlayer()
30    {
31        // 유니티의 Instantiate가 아니다. 사용법이 다르다.
32        PhotonNetwork.Instantiate("Player",
33                                new Vector3(0, 1, 0),
34                                Quaternion.identity,
35                                0);
36
37        yield return null;
38    }
39
40    }
41
42    }
43
44    }
45
46    }
47
48    }
49
50    }
51
52    }
```

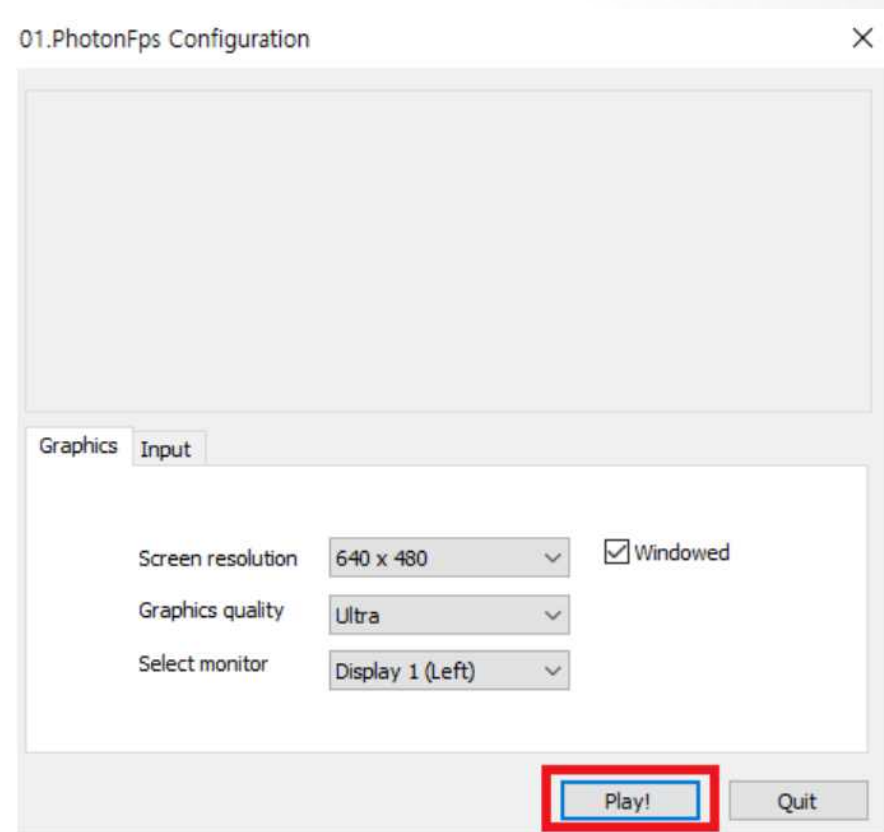
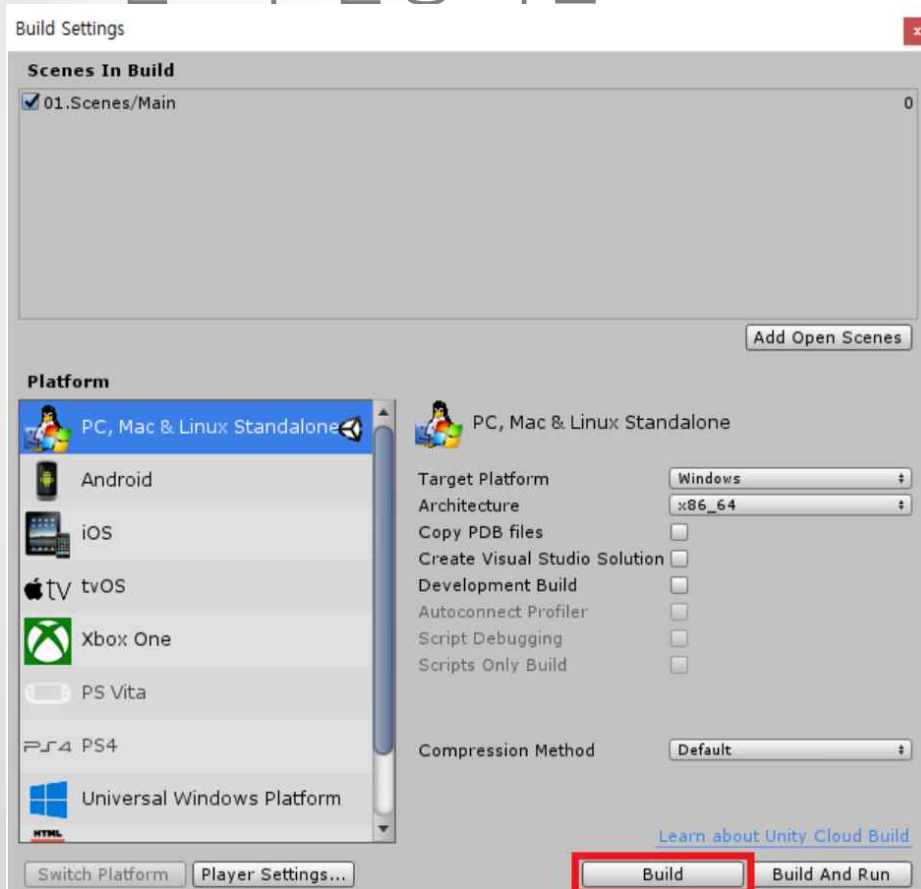






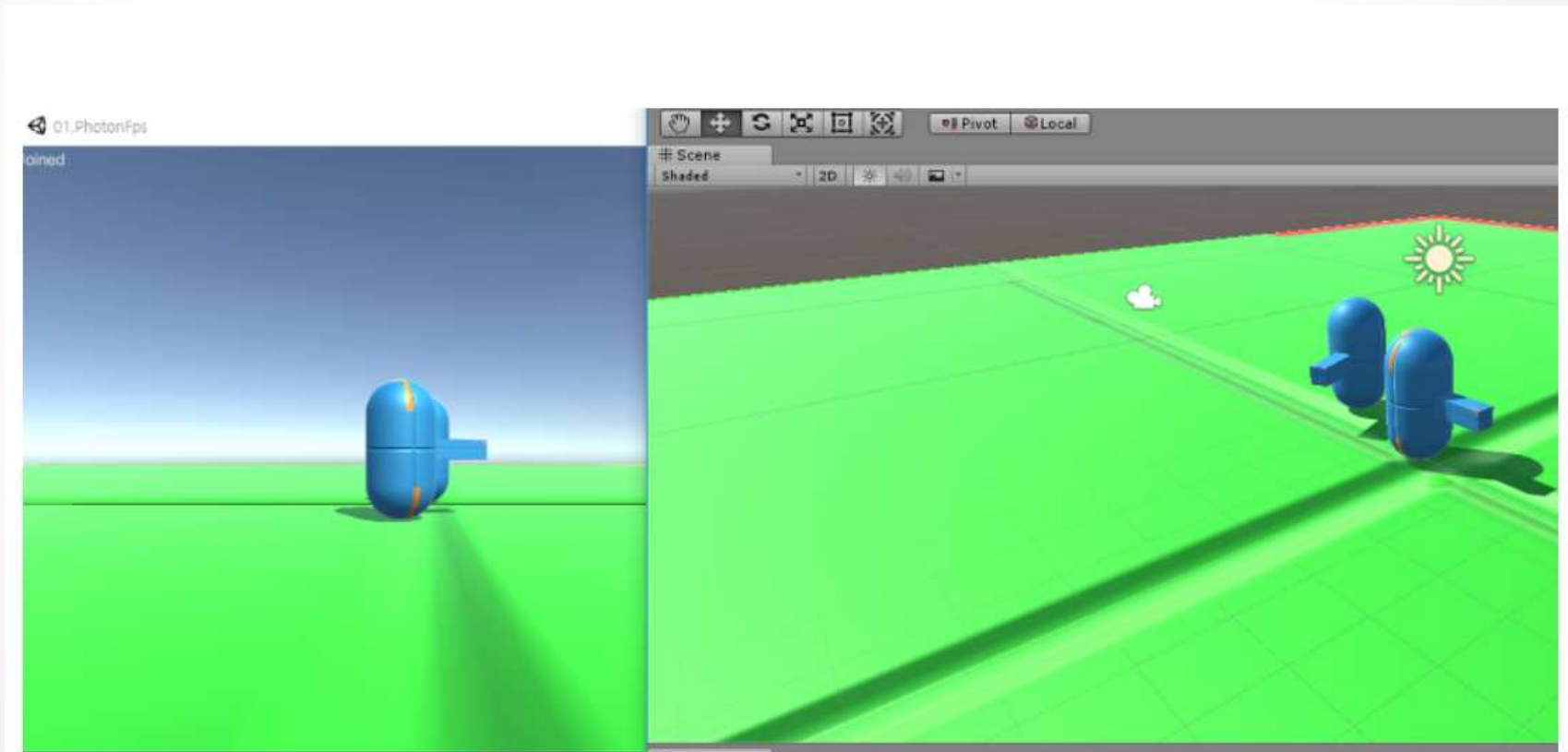
# 포톤네트워크를 사용한 FPS 개발

- 빌드후 실행 확인



# 포톤네트워크를 사용한 FPS 개발

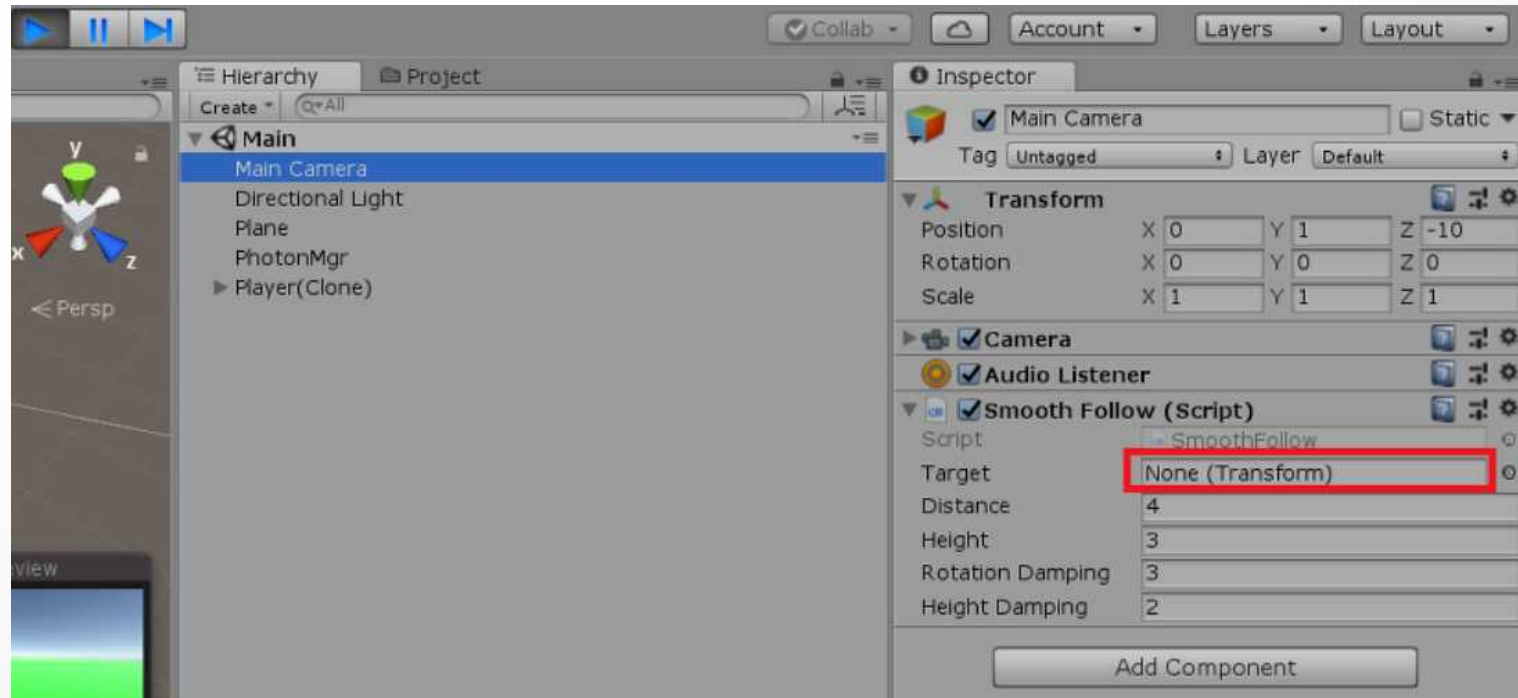
- 빌드후 실행 확인



# 포톤네트워크를 사용한 FPS 개발

- 메인 카메라 설정

- 게임 실행을 해보니 Target에 오브젝트가 없음
- SmoothFollow 스크립트에 target이 private인 것이 문제



# 포톤네트워크를 사용한 FPS 개발

- 메인 카메라 설정

- 게임 실행을 해보니 Target에 오브젝트가 없음
- SmoothFollow 스크립트에 target이 private인 것이 문제

```
3 namespace UnityStandardAssets.Utility
4 {
5     public class SmoothFollow : MonoBehaviour
6     {
7
8         // The target we are following
9         [SerializeField]
10         public Transform target;
11         // The distance in the x-z plane to the target
12         [SerializeField]
13         private float distance = 10.0f;
```

# 포톤네트워킹을 사용한 FPS 개발

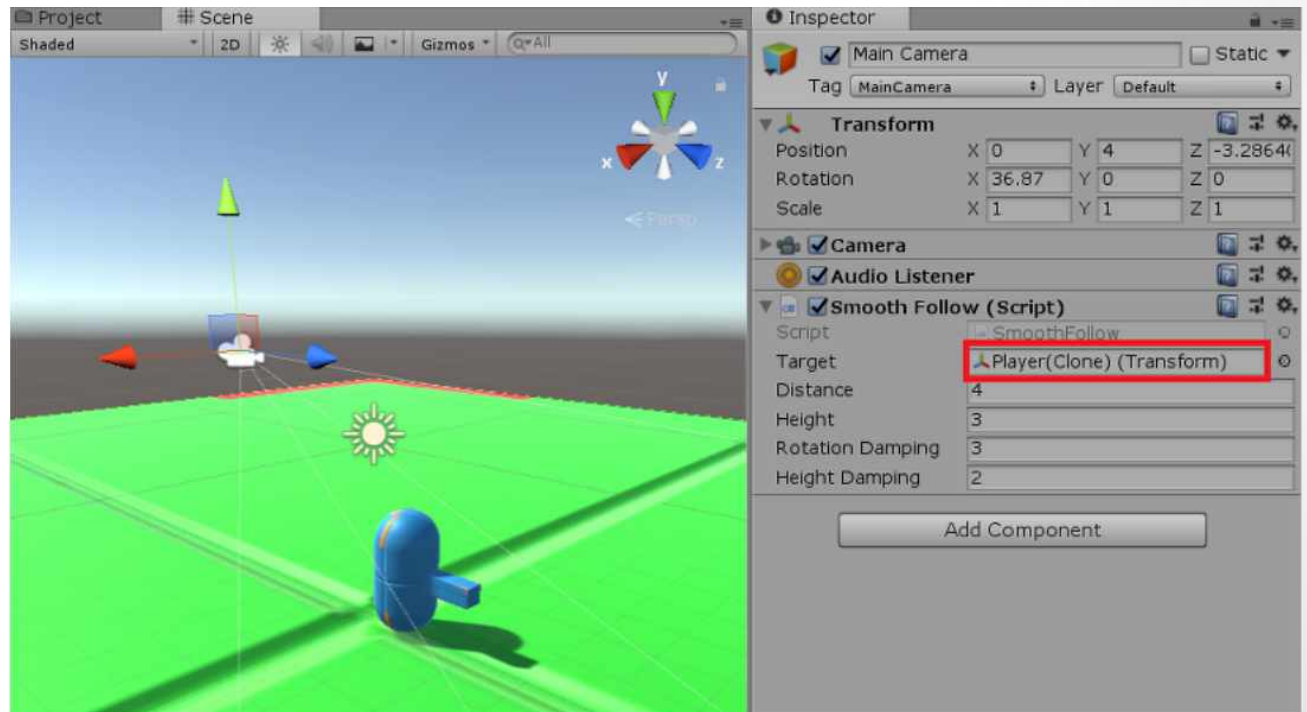
- 플레이어 스크립트 수정

- 네트워크에 들어온 사람이 아닌 오직 나만이 카메라 제어권을 가지도록 설정하기

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.StandardAssets.Utility;
5
6 public class PlayerCtl : MonoBehaviour {
7
8     public float speed = 5.0f;
9     public float rotSpeed = 120.0f;
10
11     private Transform tr;
12     private PhotonView pv;
13
14     void Start()
15     {
16         tr = GetComponent<Transform>();
17         pv = GetComponent<PhotonView>();
18
19         if (pv.isMine)
20         {
21             // 자신의 플레이어에게만 카메라 제어권을 연결한다.
22             Camera.main.GetComponent<SmoothFollow>().target = tr;
23         }
24     }
25 }
```

# 포톤네트워킹을 사용한 FPS 개발

- 플레이어 스크립트 수정
  - 네트워크에 들어온 사람이 아닌 오직 나만이 카메라 제어권을 가지도록 설정하기



# 포톤네트워드를 사용한 FPS 개발

- 플레이어 캐릭터 구분
  - 머터리얼 속성을 변경하여 구분 하기, isMine을 활용

