

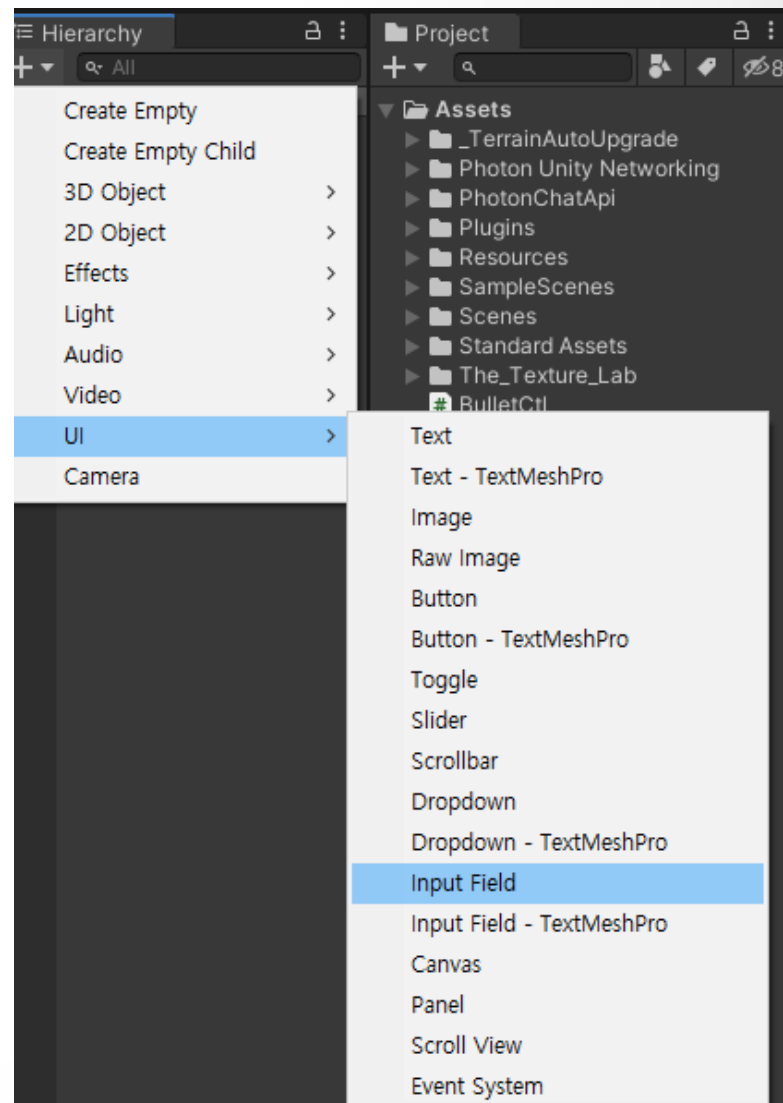
# 포톤네트워크 플레이어 정보 및 채팅

# 포톤네트워크 플레이어 정보 및 채팅

- 네트워크 게임의 확장 (사실 UI 추가 학습 및 네트워킹)
  - 사용자로부터 아이디 입력 받기
  - 채팅 메시지 전달할수 있도록 하기
  - 로비 만들고 방 따로 만들고 처리할수 있도록 하기
  - 퀘스트로 해당 기능들을 포팅하고 사용할수 있도록 하기
  - 보이스 채팅 적용하기

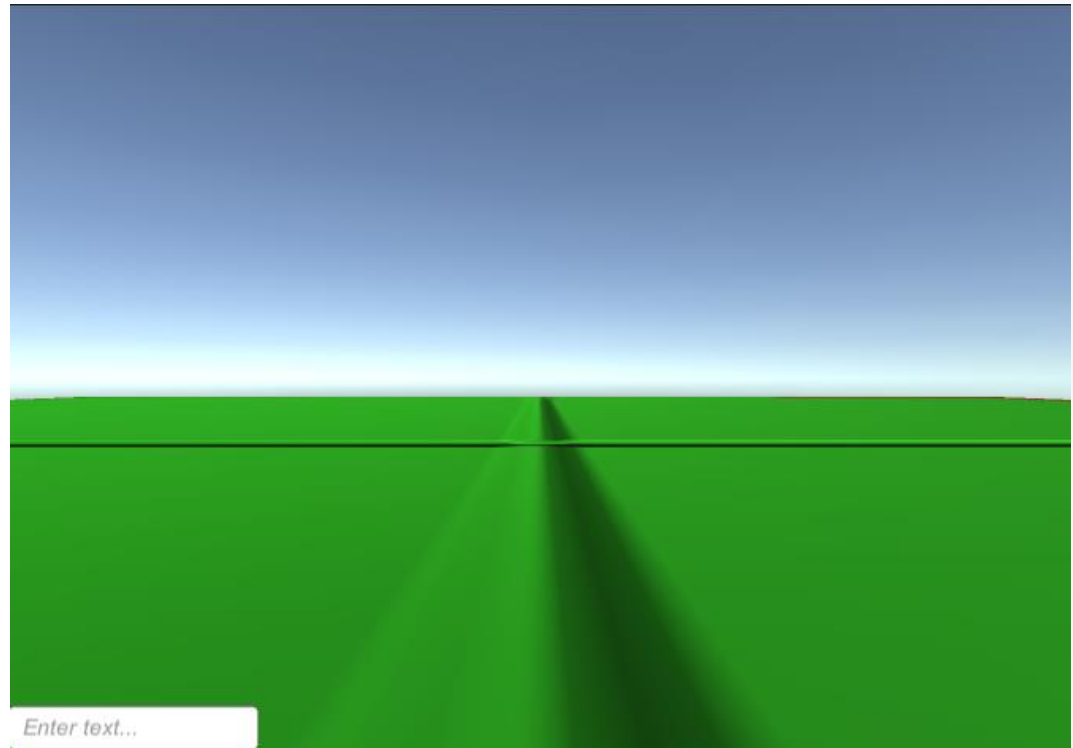
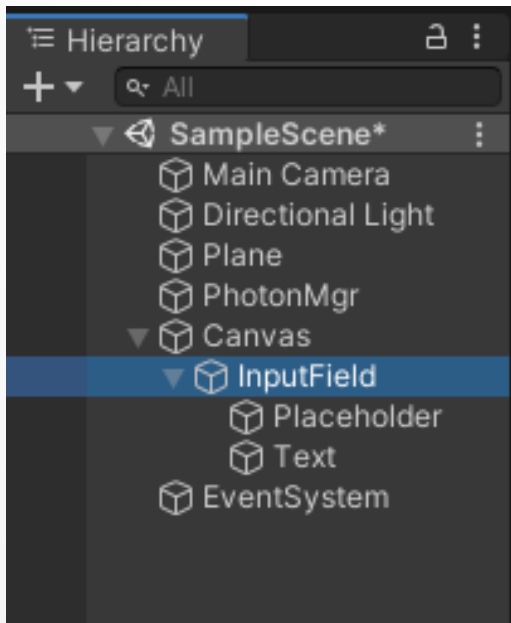
# 포톤네트워크 플레이어 정보 및 채팅

- 사용자로부터 입력을 받기
  - 지난 시간에 만든 NetworkGame 프로젝트를 열기
  - 안되었다면 클래스룸에서 받아서 처리하세요
  - UI->Input Field를 선택하면 다음과 같은 UI가 생성됨



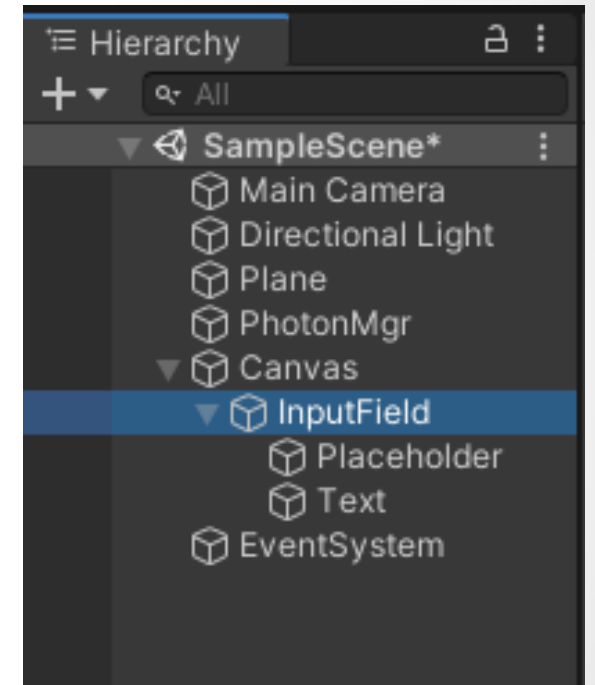
# 포톤네트워크 플레이어 정보 및 채팅

- 사용자로부터 입력을 받기
  - 지난 시간에 만든 NetworkGame 프로젝트를 열기
  - UI->Input Field를 선택하면 다음과 같은 UI가 생성됨



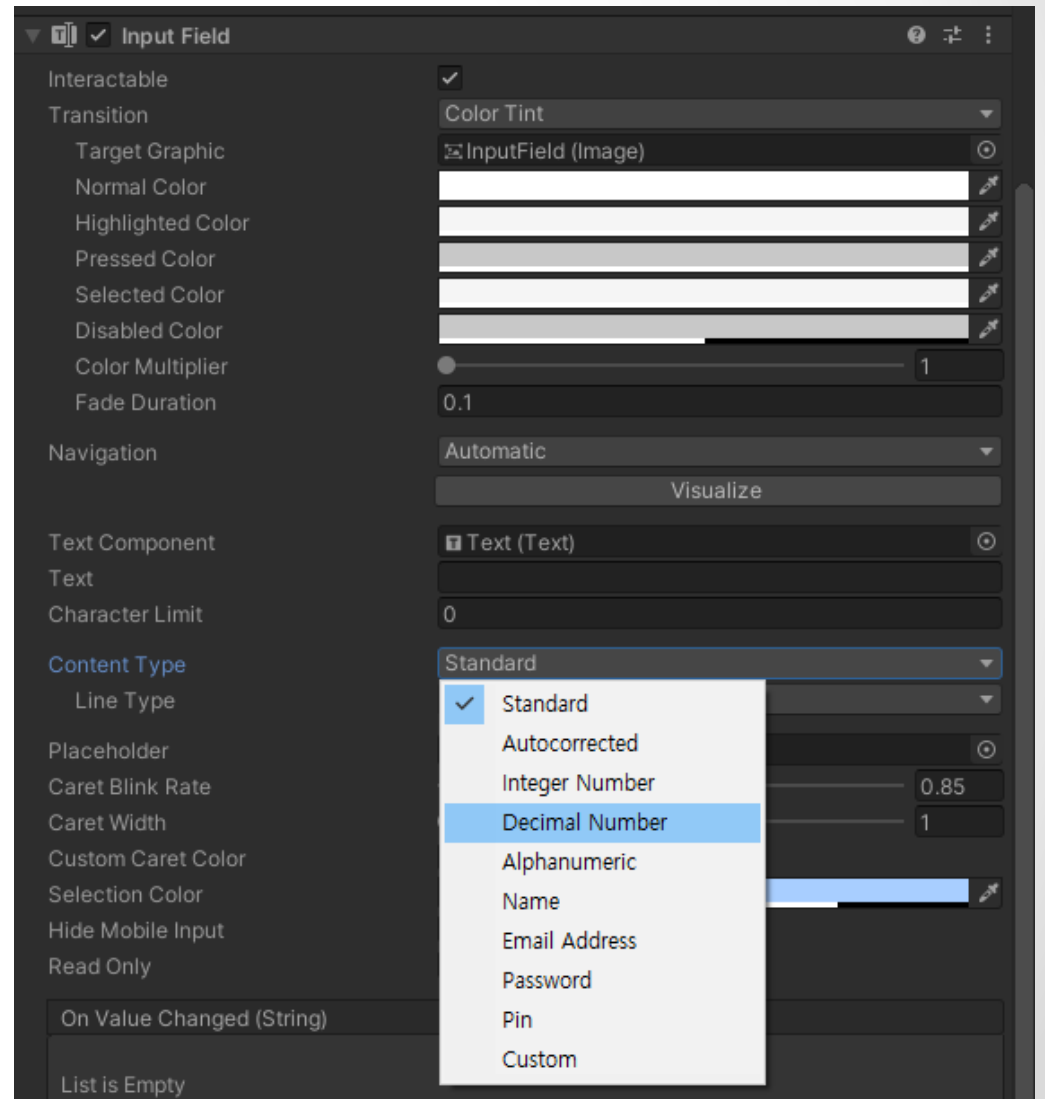
# 포톤네트워크 플레이어 정보 및 채팅

- InputField
  - 사용자로부터 키보드 입력을 받는 객체
- Placeholder
  - 사용자에게 무엇을 입력할건지를 알려줌 (도움말)
  - 사용자가 텍스트를 입력하면 없어짐
- Text
  - 사용자가 입력하게 되는 텍스트창



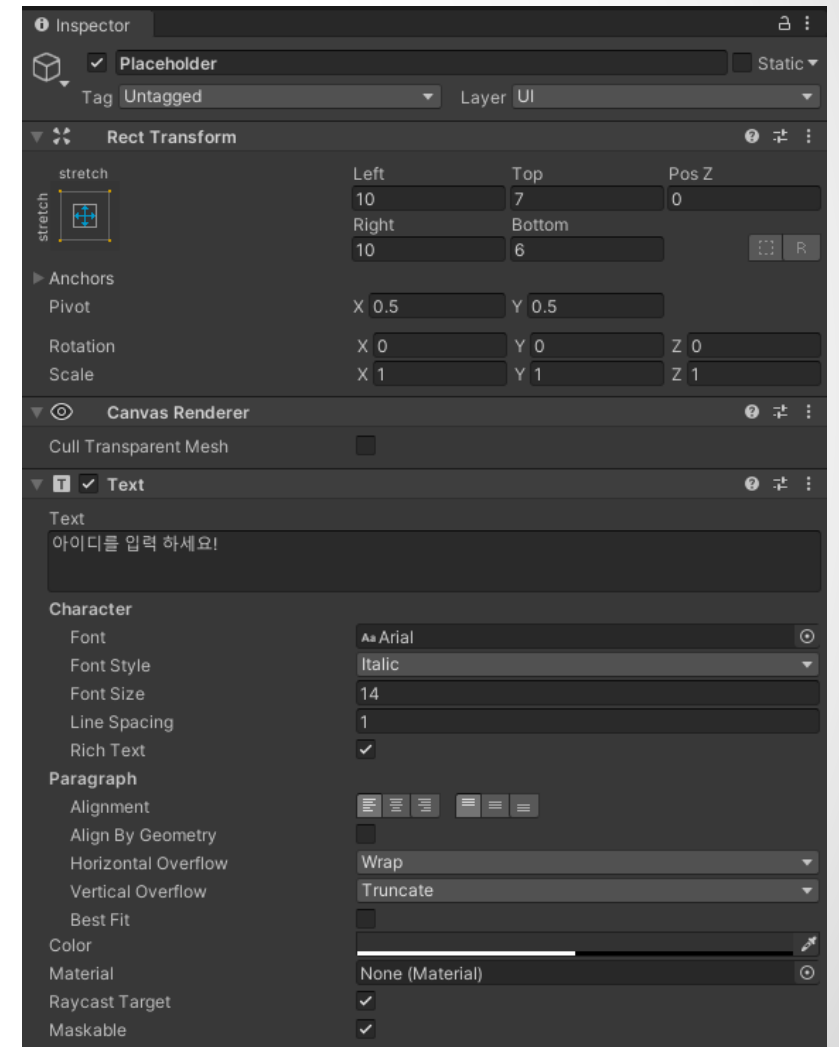
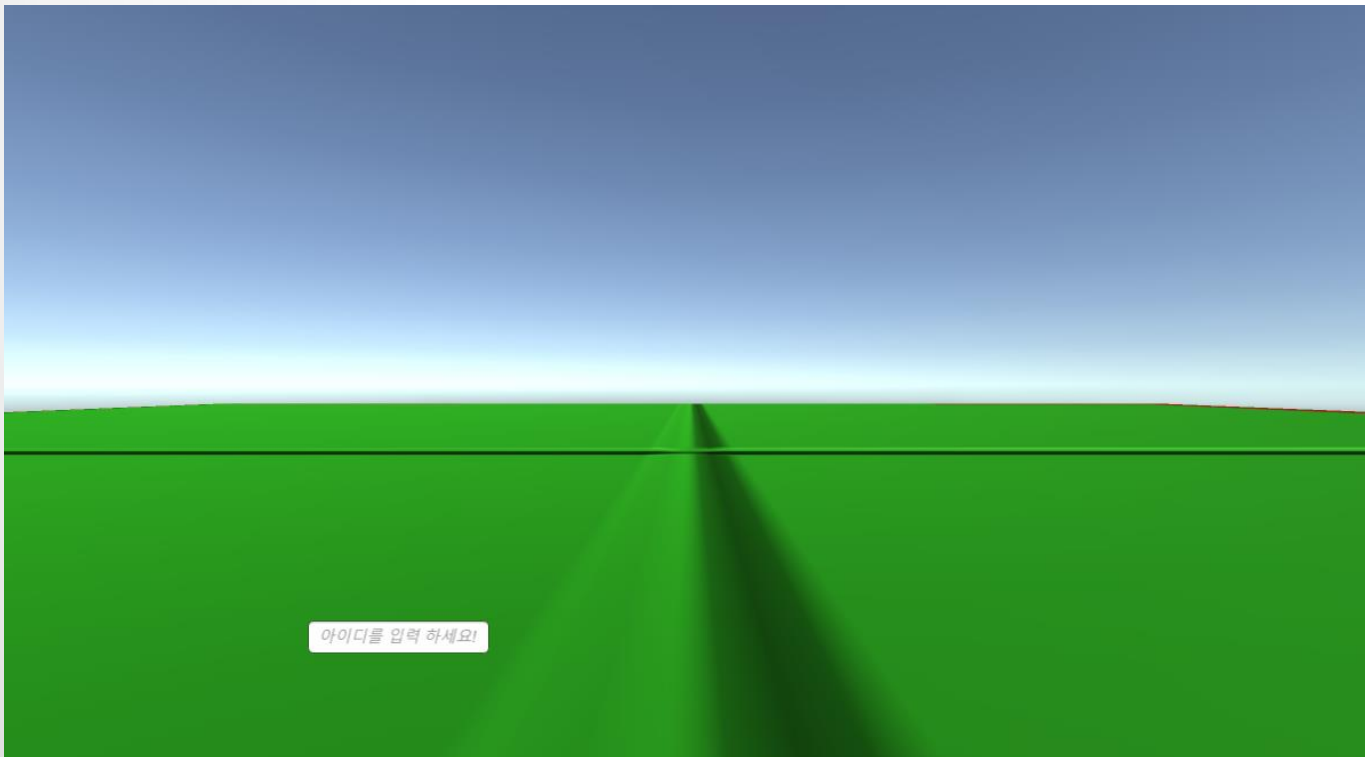
# 포톤네트워크 플레이어 정보 및 채팅

- InputField 의 ContentType은 다양한 타입을 제공함



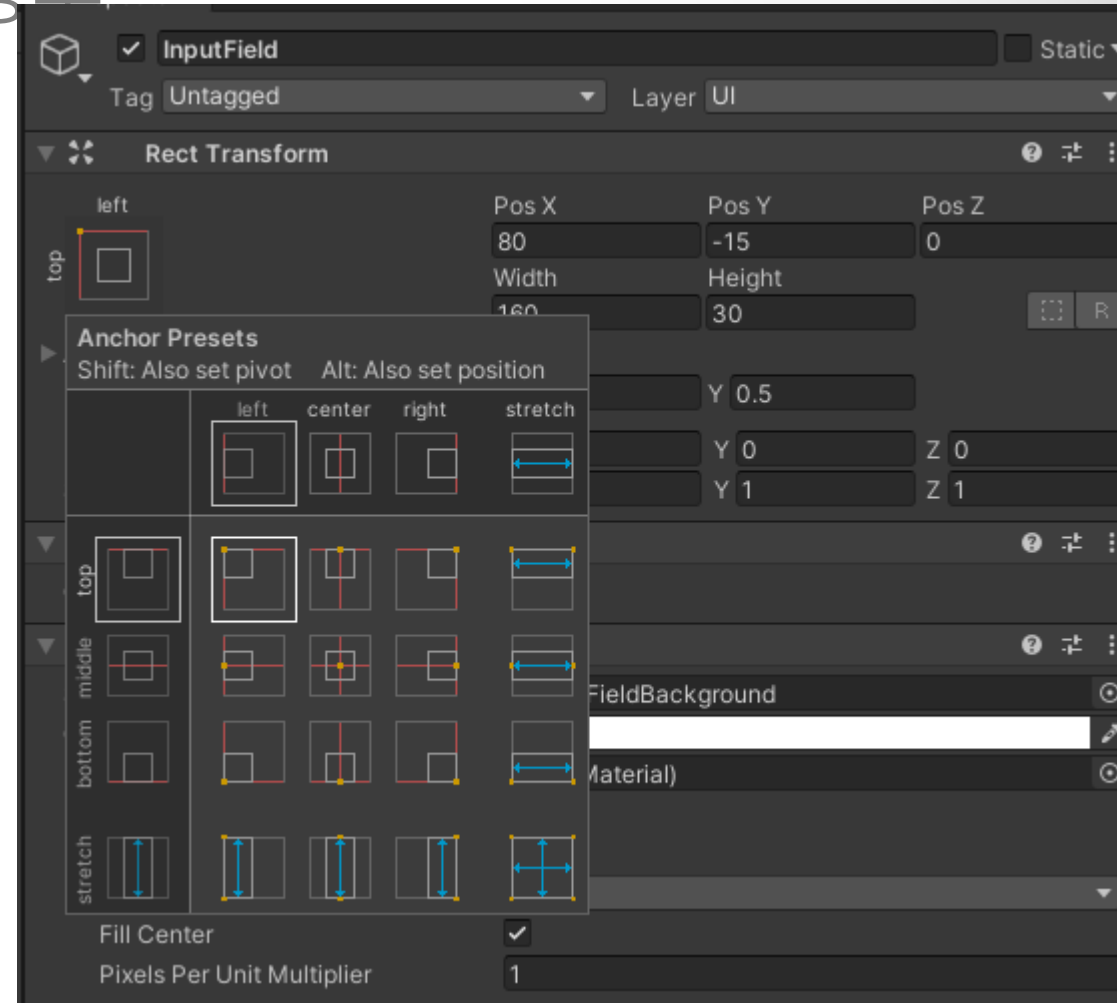
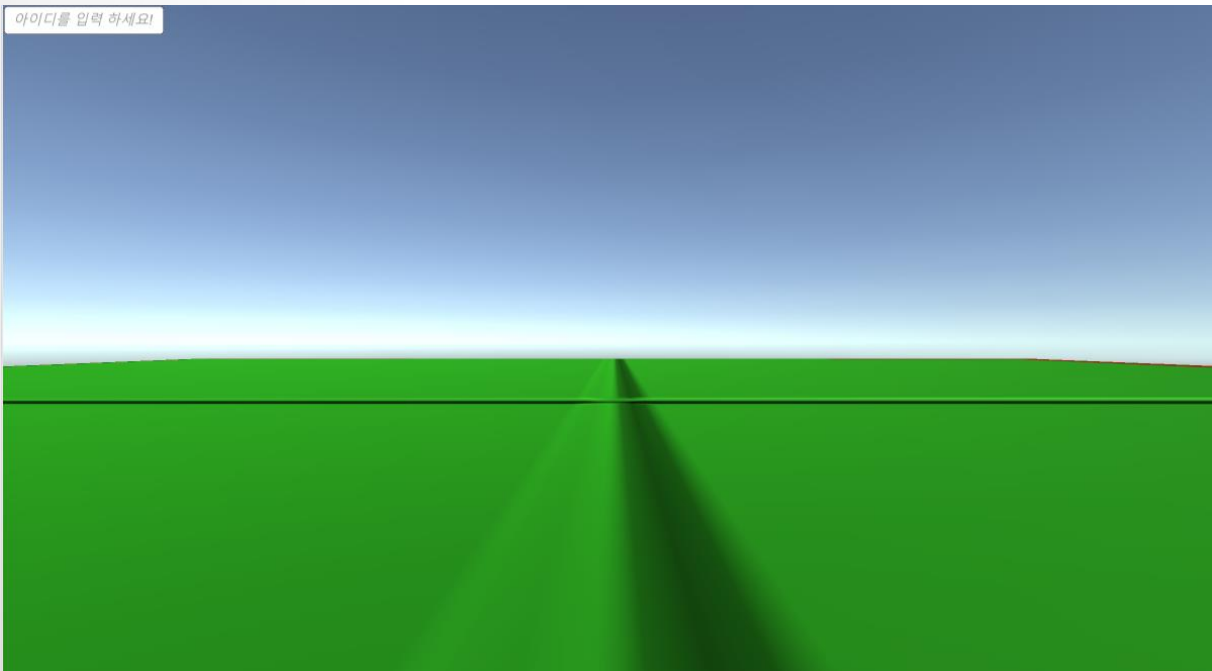
# 포톤네트워크 플레이어 정보 및 채팅

- Placeholder 사용하기!
  - 아이디를 입력하세요! 를 텍스트에 입력해보기!
  - 실행후에 직접 아이디를 입력해서 없어지는것 확인하기!



# 포톤네트워크 플레이어 정보 및 채팅

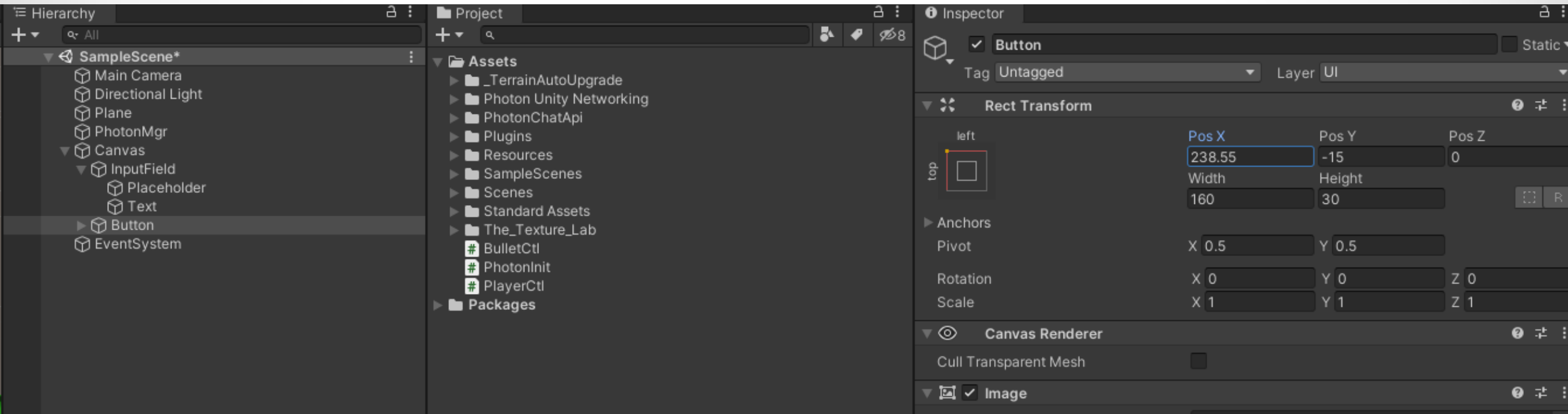
- InputField의 위치를 앵커 프리셋 실행
- Alt 키를 누르고 왼쪽 최상된 위치





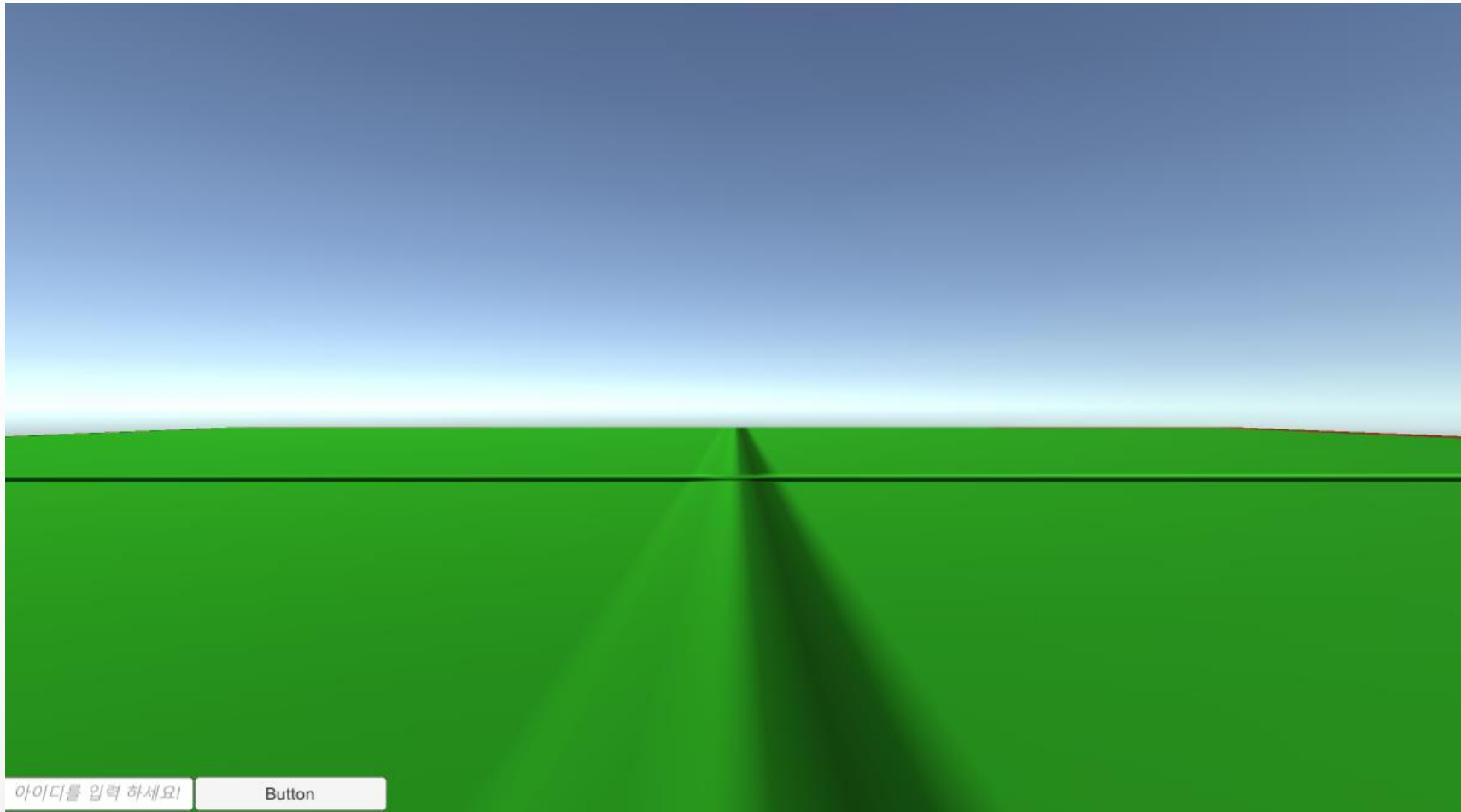
# 포톤네트워크 플레이어 정보 및 채팅

- Button 생성후 다음과 같이 배치하기



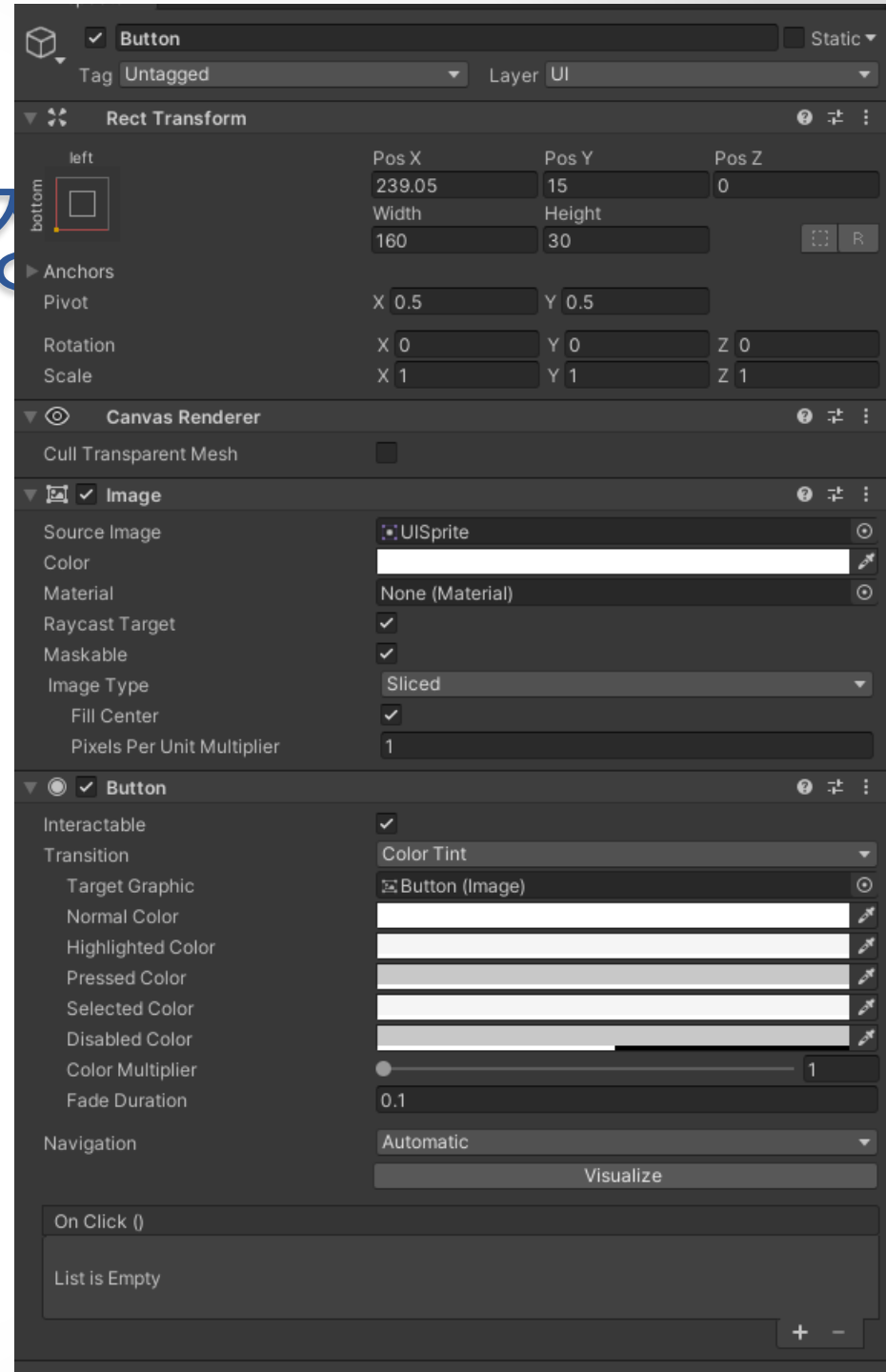
# 포톤네트워크 플레이어 정보 및 채팅

- 버튼 생성결과 확인 (아래로 내려주는게 좋습니다)



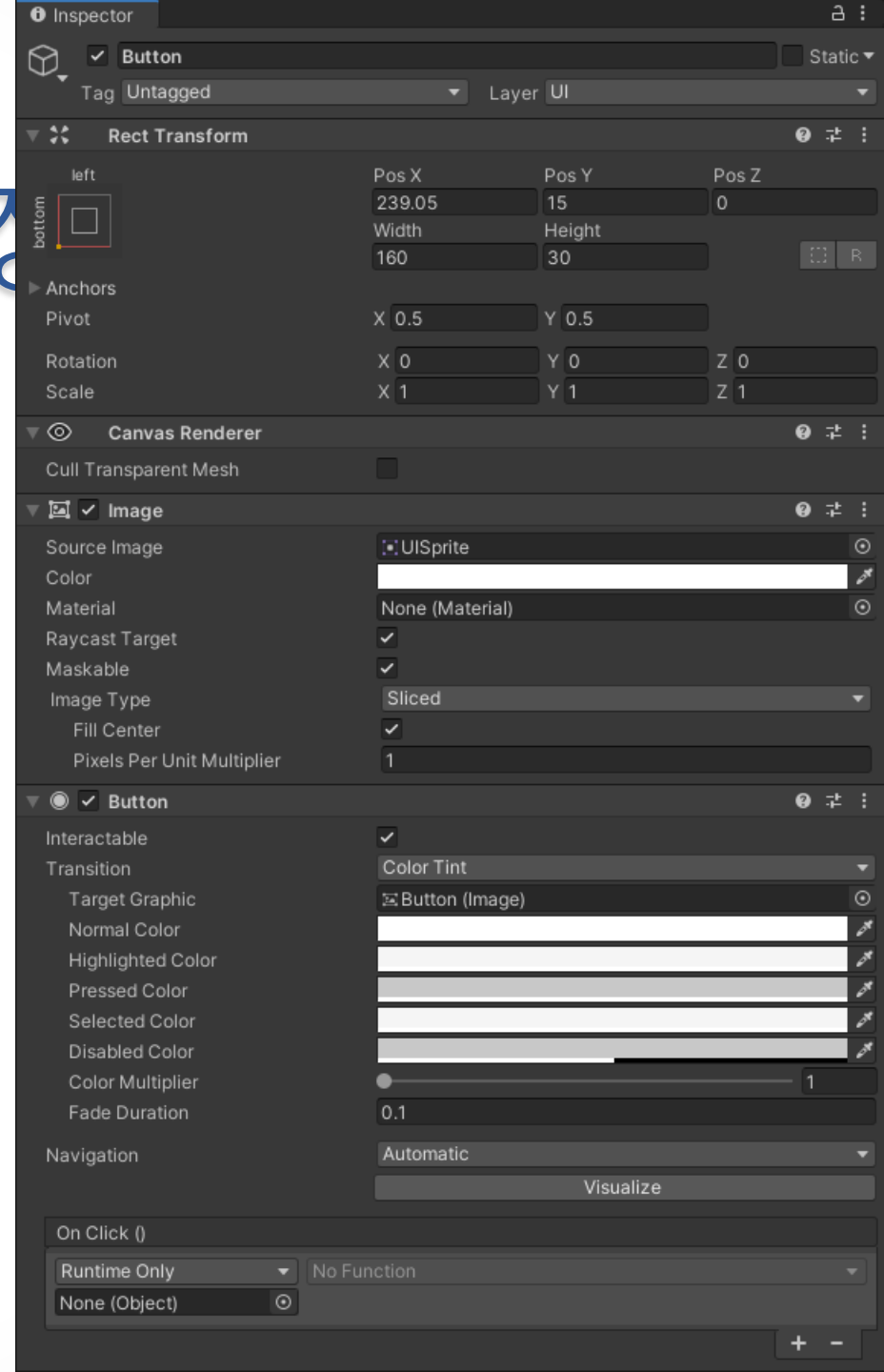
# 포톤네트워크 플레이어 강

- Button 선택후 OnClick 에서 + 클릭



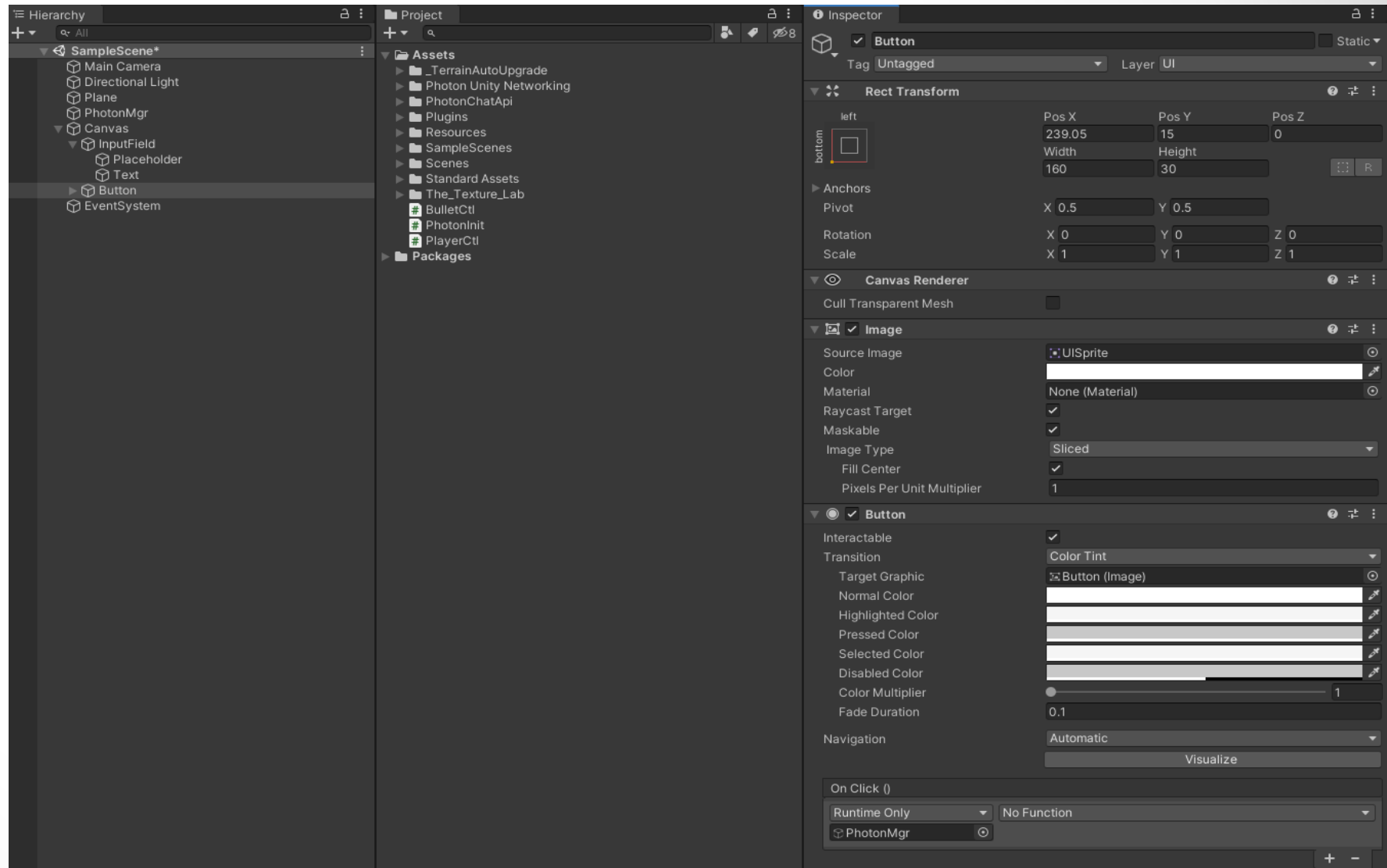
# 포톤네트워크 플레이어

- Button 선택후 OnClick 에서 + 클릭



# 포톤네트워크 플레이어 정보 및 채팅

- 버튼 onClick 함수추가



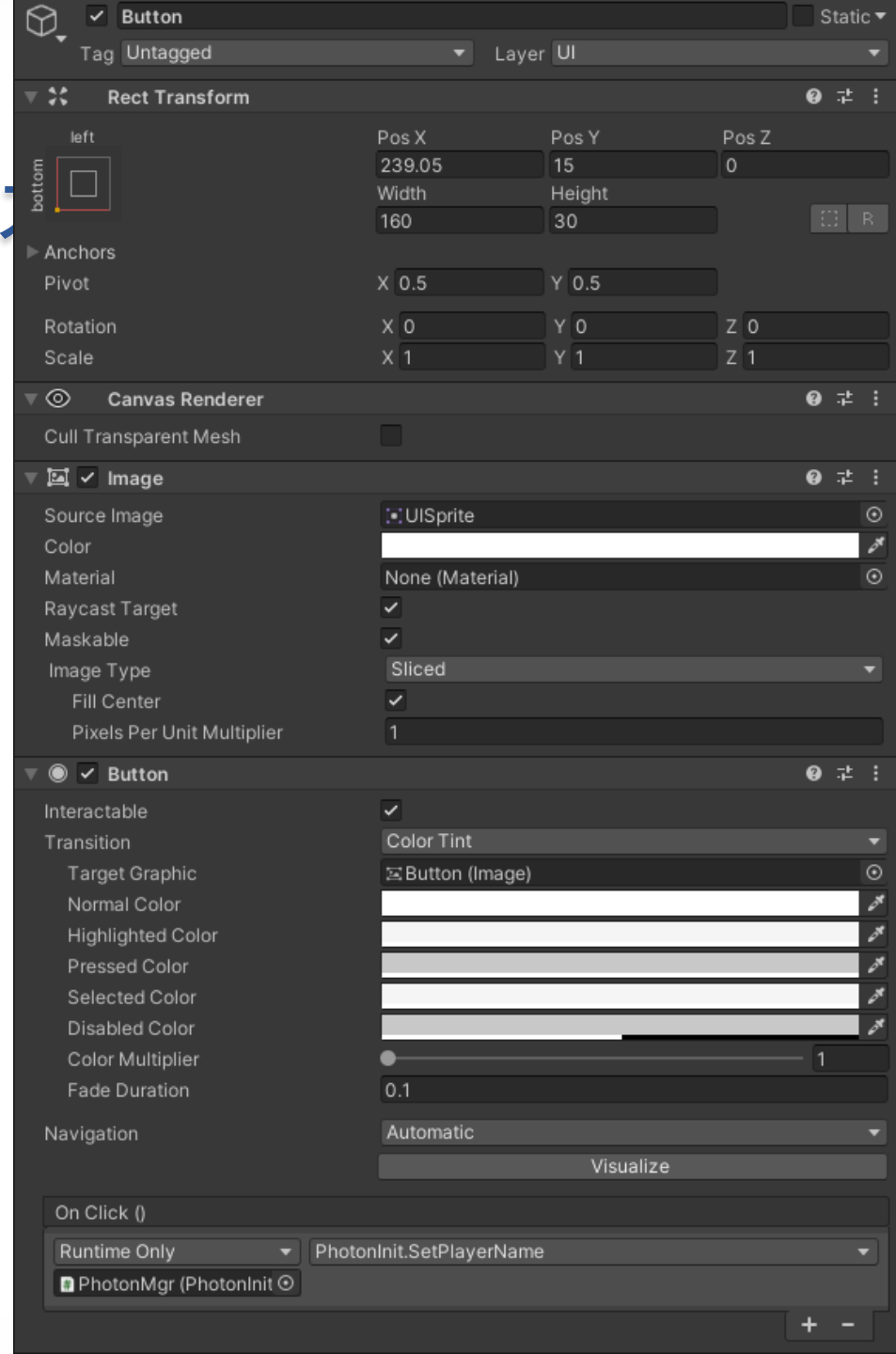
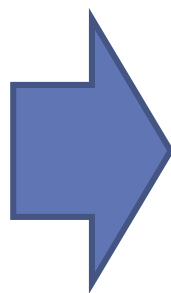
# 포톤네트워크 플레이어

- PhotonInit.cs 파일 수정후 UI 설정하기

```
PhotonInit.cs
Assembly-CSharp PhotonInit

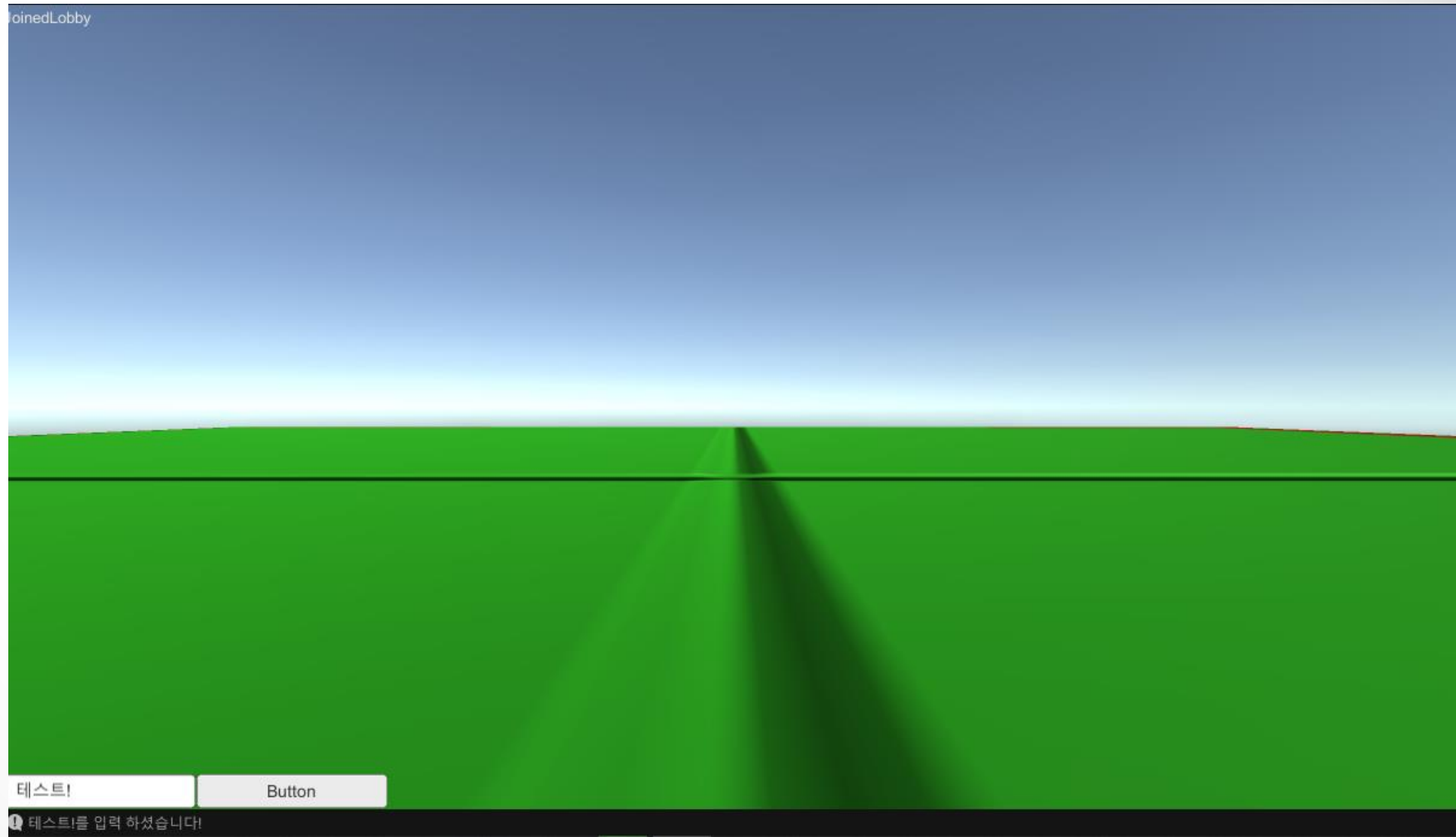
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon;
5 using UnityEngine.UI;
6
7 public class PhotonInit : Photon.PunBehaviour
8 {
9     public InputField playerInput;
10
```

```
public void SetPlayerName()
{
    Debug.Log(playerInput.text + "를 입력 하셨습니다!");
}
```



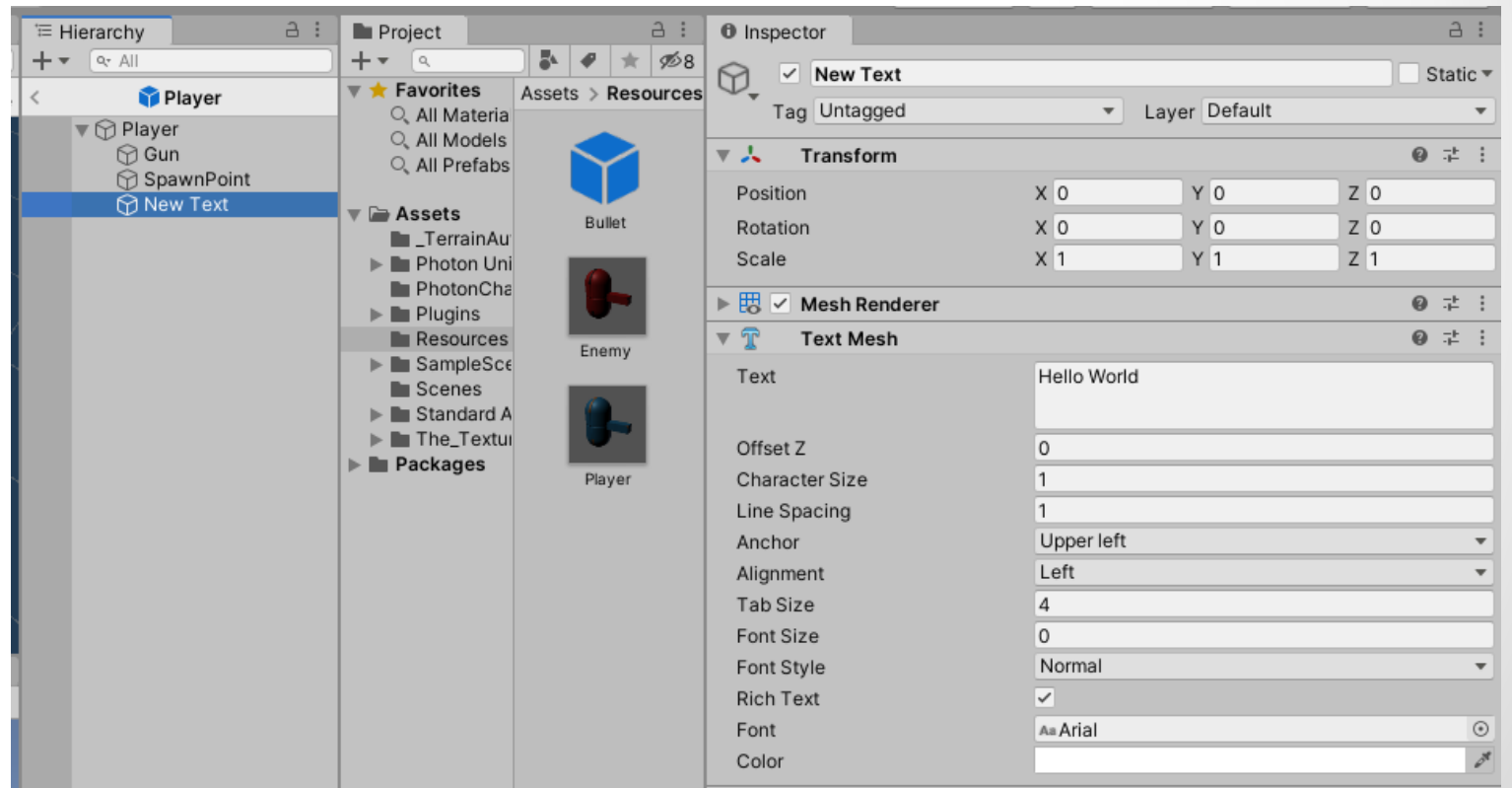
# 포톤네트워크 플레이어 정보 및 채팅

- 결과 확인



# 포톤네트워크 플레이어 정보 및 채팅

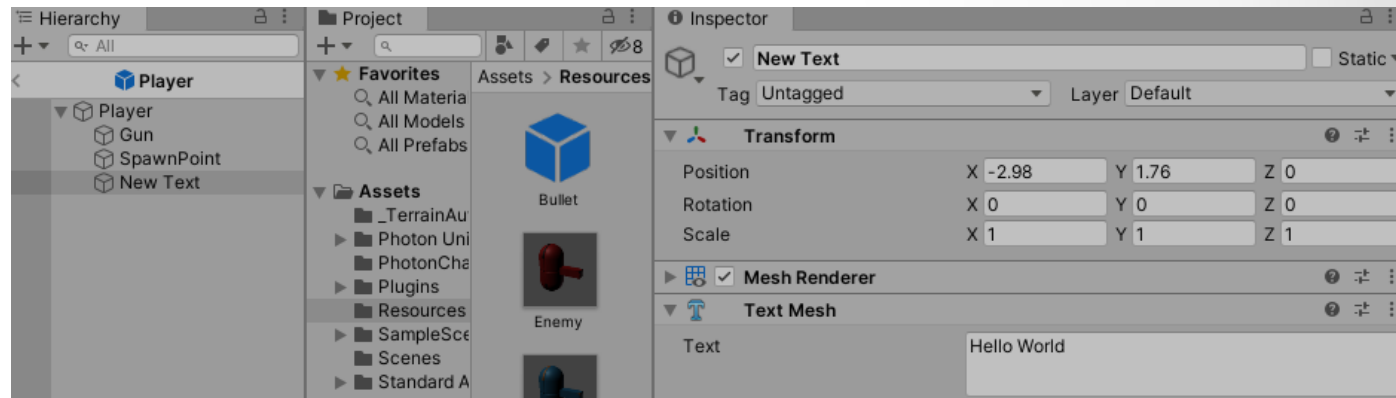
- TextMesh 객체를 추가 (3차원 Text 처리를 담당!)





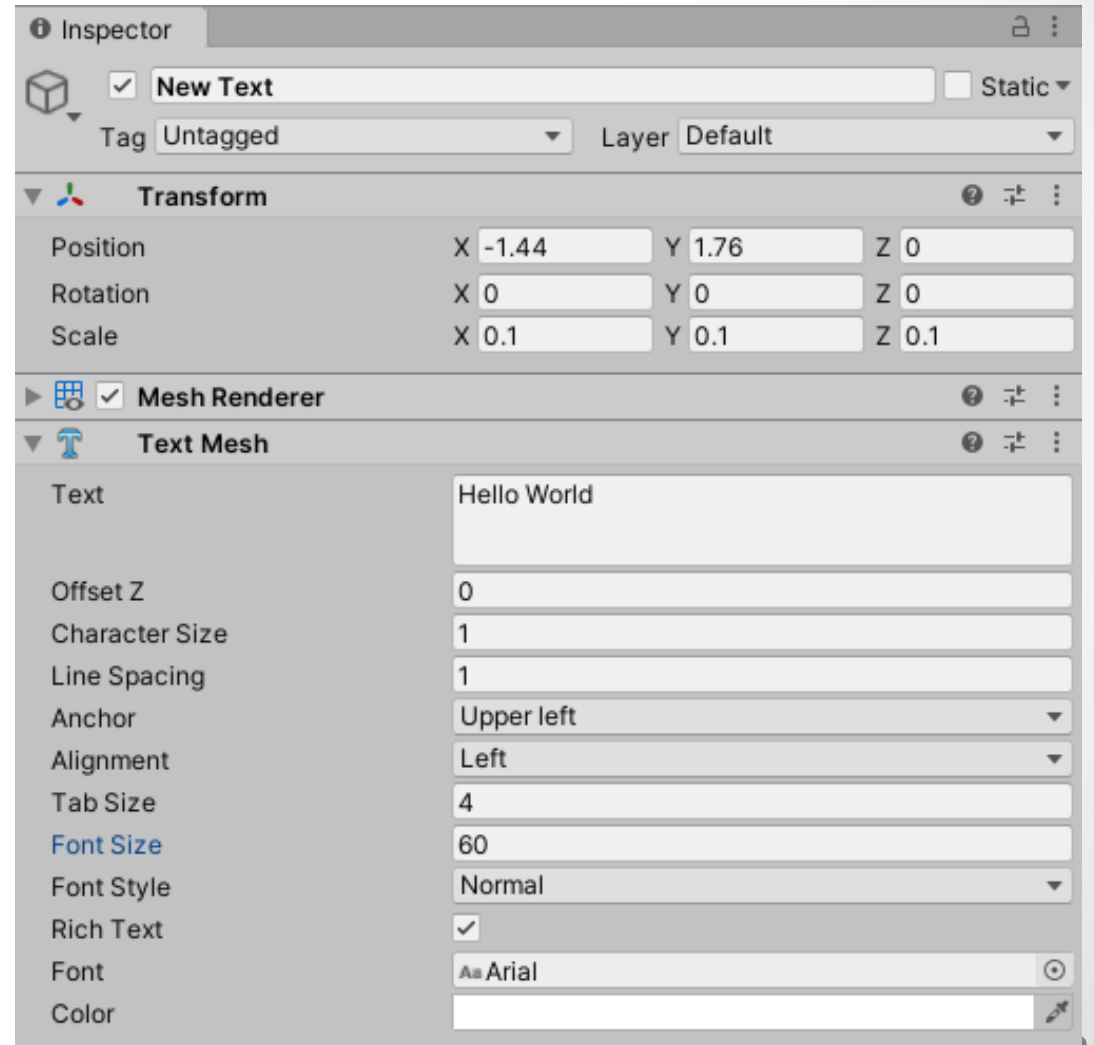
# 포톤네트워크 플레이어 정보 및 채팅

- 텍스트의 위치 변경후 결과 확인!
  - 텍스트 화질이 너무 떨어짐, 어떻게 개선할수 있을까?



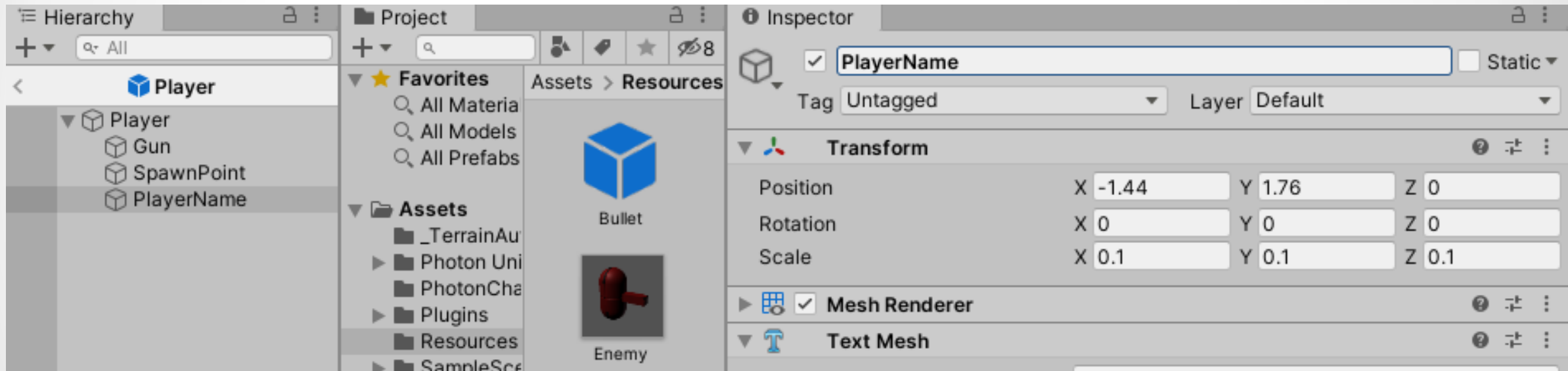
# 포톤네트워크 플레이어 정보 및 채팅

- 3D Text의 해상도를 높이려면?
- 폰트사이즈 높이고 스케일 값은 줄여야함



# 포톤네트워크 플레이어 정보 및 채팅

- 3DText 이름을 PlayerName으로 변경하기



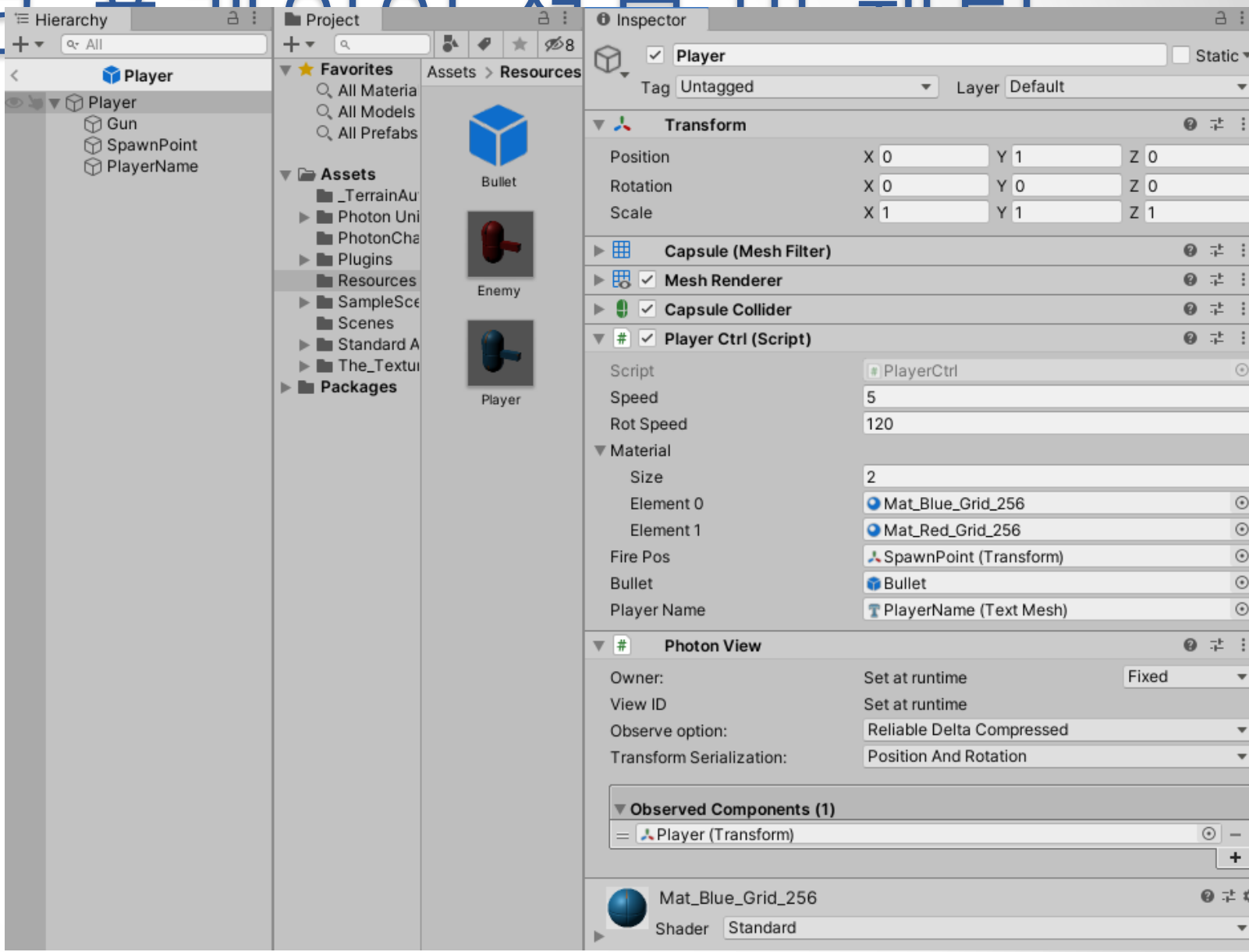
# 포톤네트워크 플레이어 정보 및 채팅

- PlayerCtrl.cs 파일 수정!
  - 멤버 변수에 TextMesh 를 받을 PlayerName추가
  - 이름을 관리할 변수인 name 추가

```
5
6 public class PlayerCtrl : MonoBehaviour
7 {
8     public float speed = 5.0f;
9     public float rotSpeed = 120.0f;
10    private Transform tr;
11    private PhotonView pv;
12    public Material[] _material;
13
14    private Vector3 currPos;
15    private Quaternion currRot;
16
17    public Transform firePos;
18    public GameObject bullet;
19
20    private bool isDie = false;
21    private int hp = 100;
22    private float respawnTime = 3.0f;
23
24    public TextMesh playerName;
25    string name="";
26
```

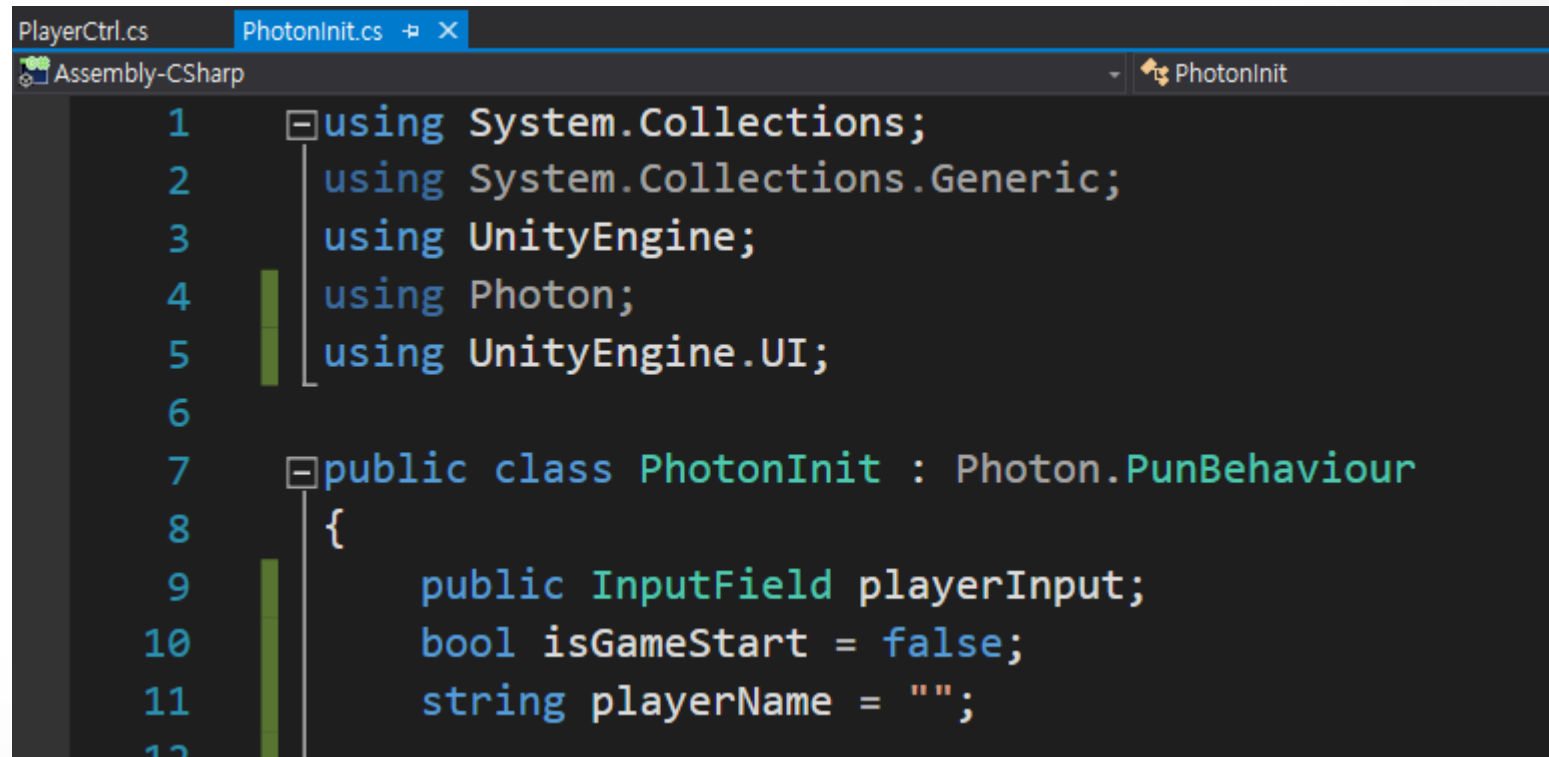
# 포톤네트워크 포게이이 정보 미 캐티

- PlayerCtrl 의 PlayerName 에 TextMesh 설정하기!  
(TextMesh의 이름도 PlayerName으로 변경)
- 프리팹의 Player를 선택하여 설정해야 함



# 포톤네트워크 플레이어 정보 및 채팅

- PlayerInit.cs 수정
  - 이제 사용자가 아이디를 입력하고 버튼을 클릭해야지만 플레이어를 생성 할수 있도록 처리한다!
  - IsGameStart, 게임 시작을 결정하는 변수
  - 입력 받은 이름을 playerName 으로 관리!



```
PlayerCtrl.cs PhotonInit.cs ✕
Assembly-CSharp PhotonInit
1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4      using Photon;
5      using UnityEngine.UI;
6
7  public class PhotonInit : Photon.PunBehaviour
8  {
9      public InputField playerInput;
10     bool isGameStart = false;
11     string playerName = "";
12
```

# 포톤네트워크 플레이어 정보 및 채팅

- PlayerInit 수정
  - 이제 사용자가 아이디를 입력하고 버튼을 클릭해야지만 플레이어를 생성 할수 있도록 처리한다!
  - IsGameStart, 게임 시작을 결정하는 변수
  - 입력 받은 이름을 PlayerName 으로 관리!

```
73     public void SetPlayerName()  
74     {  
75         Debug.Log(playerInput.text + "를 입력 하셨습니다!");  
76  
77         playerName = playerInput.text;  
78         isGameStart = true;  
79     }
```

# 포톤네트워크 플레이어 정보 및 채팅

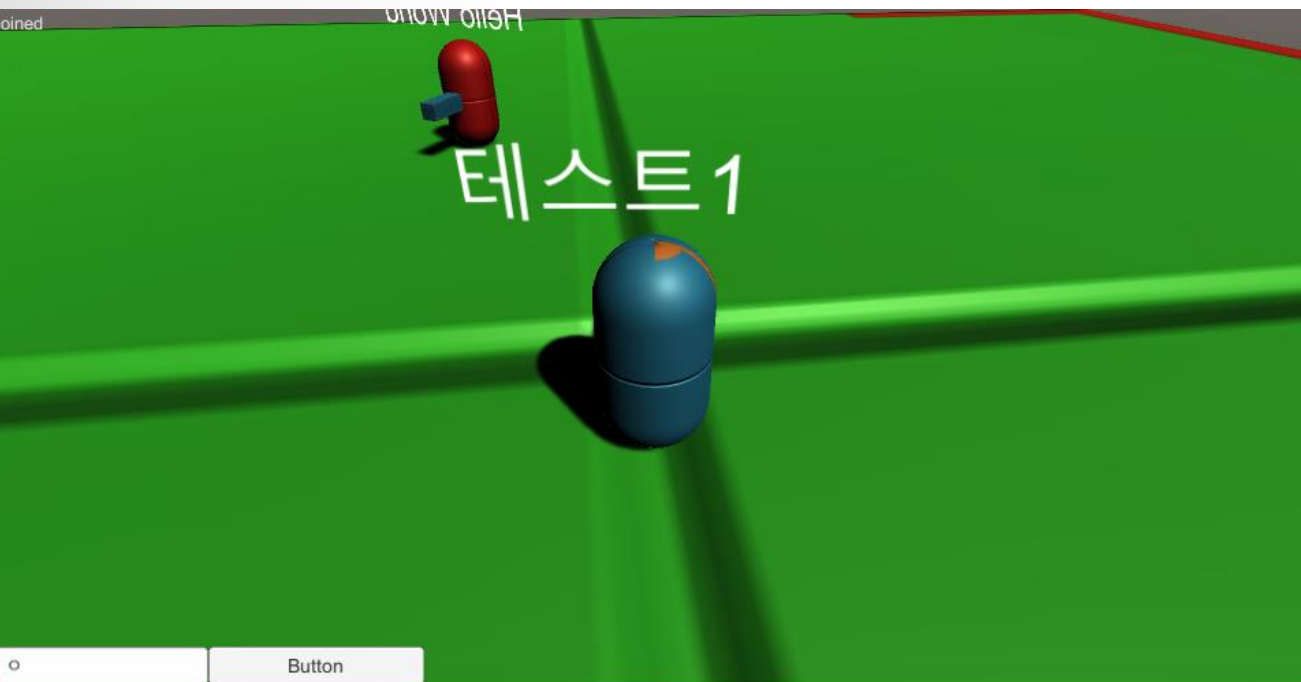
- CreatePlayer 수정
  - 기존에 바로 인스턴스를 만드는게 아니라, 사용자가 아이디를 입력하기 전까지 (isGameStart 가 true가 되기전까지) 기다리다가 (0.5 초씩 슬립), 사용자가 아이디를 입력하면, 플레이어를 생성후 플레이어 컨트롤의 이름을 변경

```
50  IEnumerator CreatePlayer()  
51  {  
52      while(!isGameStart)  
53      {  
54          yield return new WaitForSeconds(0.5f);  
55      }  
56  
57      GameObject tempPlayer = PhotonNetwork.Instantiate("Player",  
58          new Vector3(0, 1, 0),  
59          Quaternion.identity,  
60          0);  
61  
62      tempPlayer.GetComponent<PlayerCtrl>().playerName.text = playerName;  
63  
64      yield return null;  
65  }
```



# 포톤네트워크 플레이어 정보 및 채팅

- 실행후 결과 확인
  - 로컬 사용자에서는 테스트1, 테스트2가 제대로 활성화됨
  - 하지만 원격 사용자 (빨간색 적)의 경우에는 기본 이름인 Hello World로 뜨게됨



# 포톤네트워크 플레이어 정보 및 채팅

- PlayerCtrl.cs 파일 수정
  - SetPlayerName 함수를 추가
  - 여기 내부에서 TextMesh인 PlayerName의 변경이 이루어 지도록 하자

```
107     public void SetPlayerName(string name)
108     {
109         this.name = name;
110         GetComponent<PlayerCtrl>().playerName.text = this.name;
111
112     }
113
```

# 포톤네트워크 플레이어 정보 및 채팅

- PlayerCtrl.cs 파일 수정
  - OnPhotonSerializeView 함수 부분 수정
  - 세번째 값으로 설정된 이름을 전송하고, 받은 이름을 원격에서 받아서 이름을 컴포넌트로 설정한다!

```
70 void OnPhotonSerializeView(PhotonStream stream, PhotonMessageInfo info)
71 {
72     if(stream.isWriting)
73     {
74         stream.SendNext(tr.position);
75         stream.SendNext(tr.rotation);
76         stream.SendNext(name);
77     }
78     else
79     {
80         currPos = (Vector3)stream.ReceiveNext();
81         currRot = (Quaternion)stream.ReceiveNext();
82         name = (string)stream.ReceiveNext();
83         GetComponent<PlayerCtrl>().playerName.text = this.name;
84     }
85 }
86
87 }
```

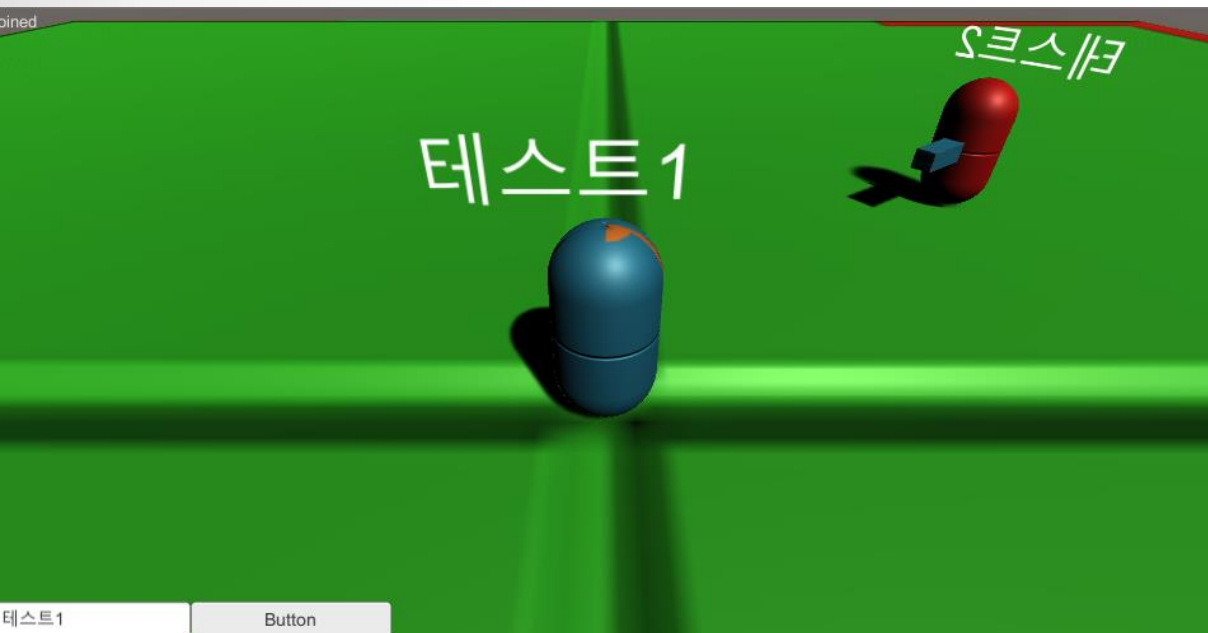
# 포톤네트워크 플레이어 정보 및 채팅

- PlayerInit.cs 파일 수정
  - CreatePlayer 에서 PlayerCtrl 클래스에 함수를 호출하도록 변경

```
50  IEnumerator CreatePlayer()  
51  {  
52      while(!isGameStart)  
53      {  
54          yield return new WaitForSeconds(0.5f);  
55      }  
56  
57      GameObject tempPlayer = PhotonNetwork.Instantiate("Player",  
58          new Vector3(0, 1, 0),  
59          Quaternion.identity,  
60          0);  
61  
62      tempPlayer.GetComponent<PlayerCtrl>().SetPlayerName(playerName);  
63  
64      yield return null;  
65  }
```

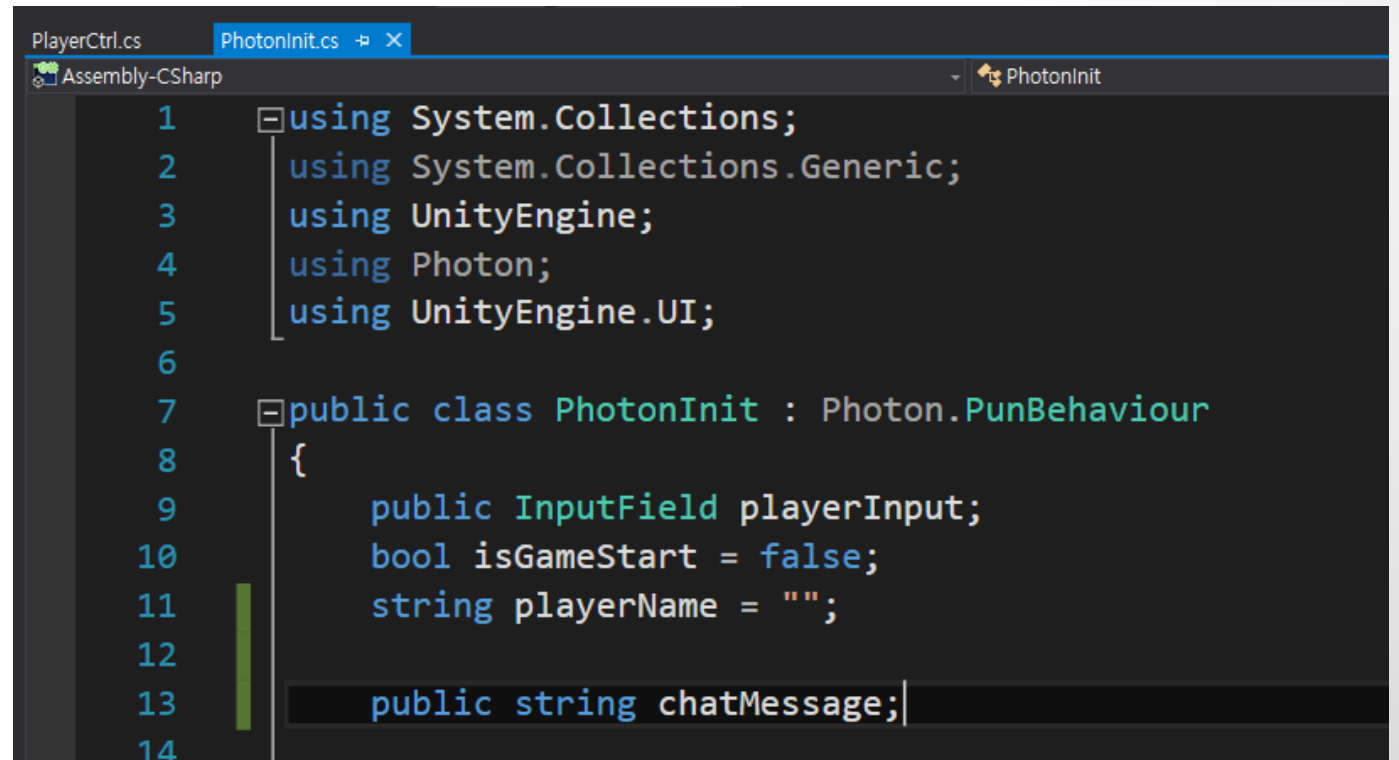
# 포톤네트워크 플레이어 정보 및 채팅

- 테스트해보기
  - 사용자가 입력이 성공하면 플레이어 객체가 만들어지고, 이름도 동기화됨
  - 이런 방법을 사용하여 나중에 DB와 연동하면 됩니다~



# 포톤네트워크 플레이어 정보 및 채팅

- 채팅 메시지 구현하기!
  - 채팅 UI 추가 (메시지를 입력 받을 InputField, 채팅 메시지를 보여줄 Scroll View를 추가해서 연결)
  - 포톤 네트워크에서는 채팅 메시지를 주고 받는 대상 (PhotonInit 이 담당, UI 기능도 같이 처리)

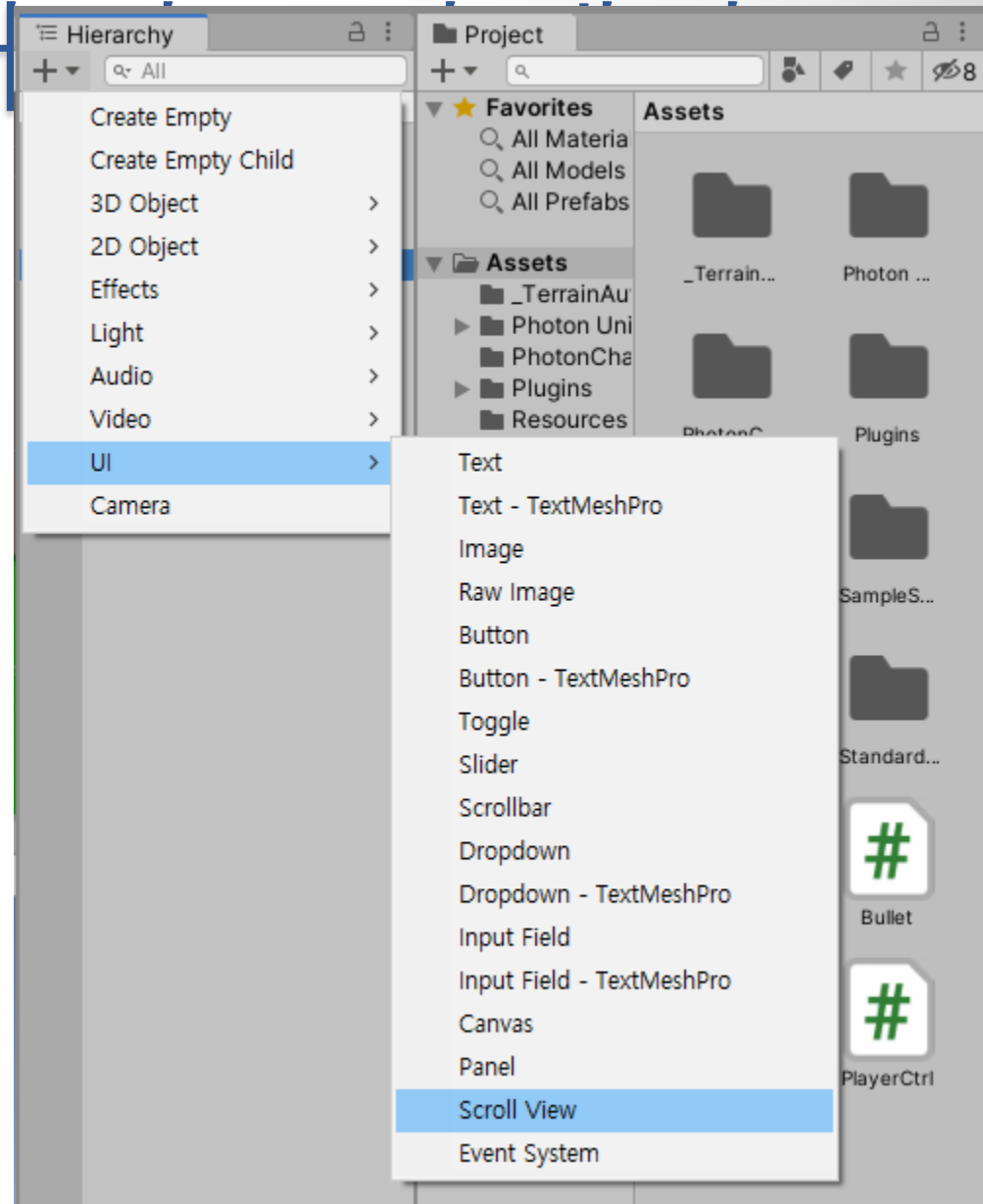


```
PlayerCtrl.cs PhotonInit.cs ×
Assembly-CSharp PhotonInit

1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4      using Photon;
5      using UnityEngine.UI;
6
7  public class PhotonInit : Photon.PunBehaviour
8  {
9      public InputField playerInput;
10     bool isGameStart = false;
11     string playerName = "";
12
13     public string chatMessage;
14
```

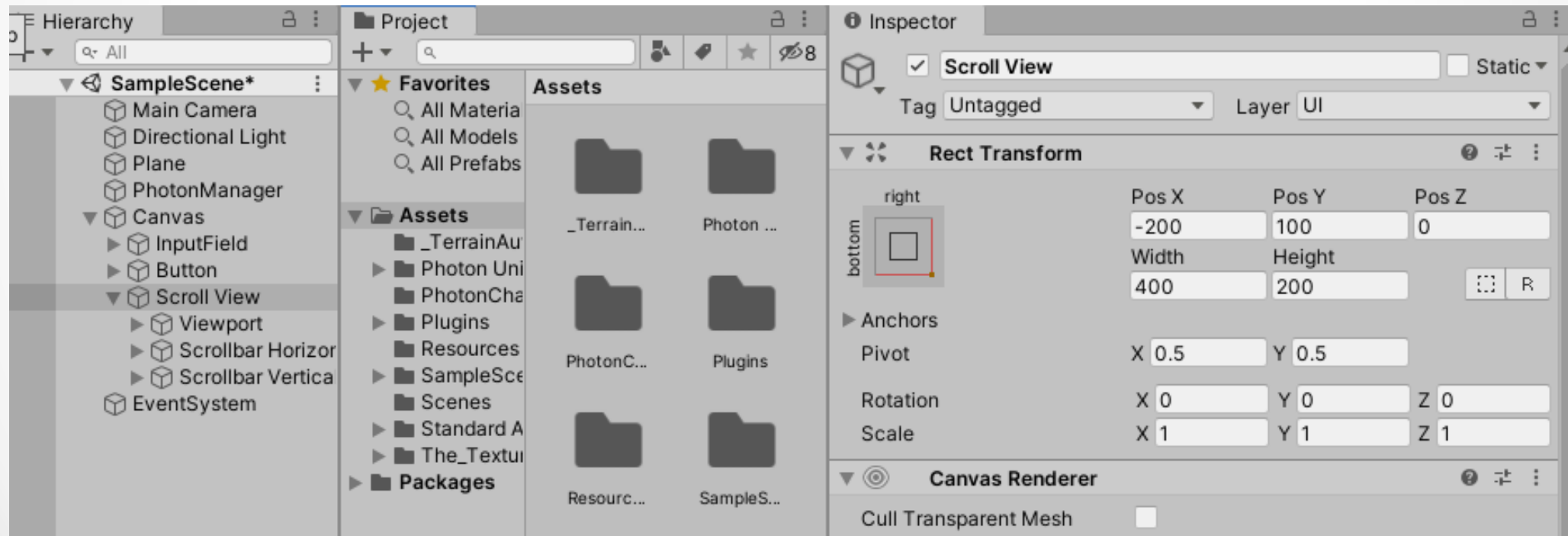
# 포톤네트워크 플레이어

- UI 중에서 계속 메시지가 쌓이는 Scroll View를 만들어 보자!  
UI->Scroll View 만들기



# 포톤네트워크 플레이어 정보 및 채팅

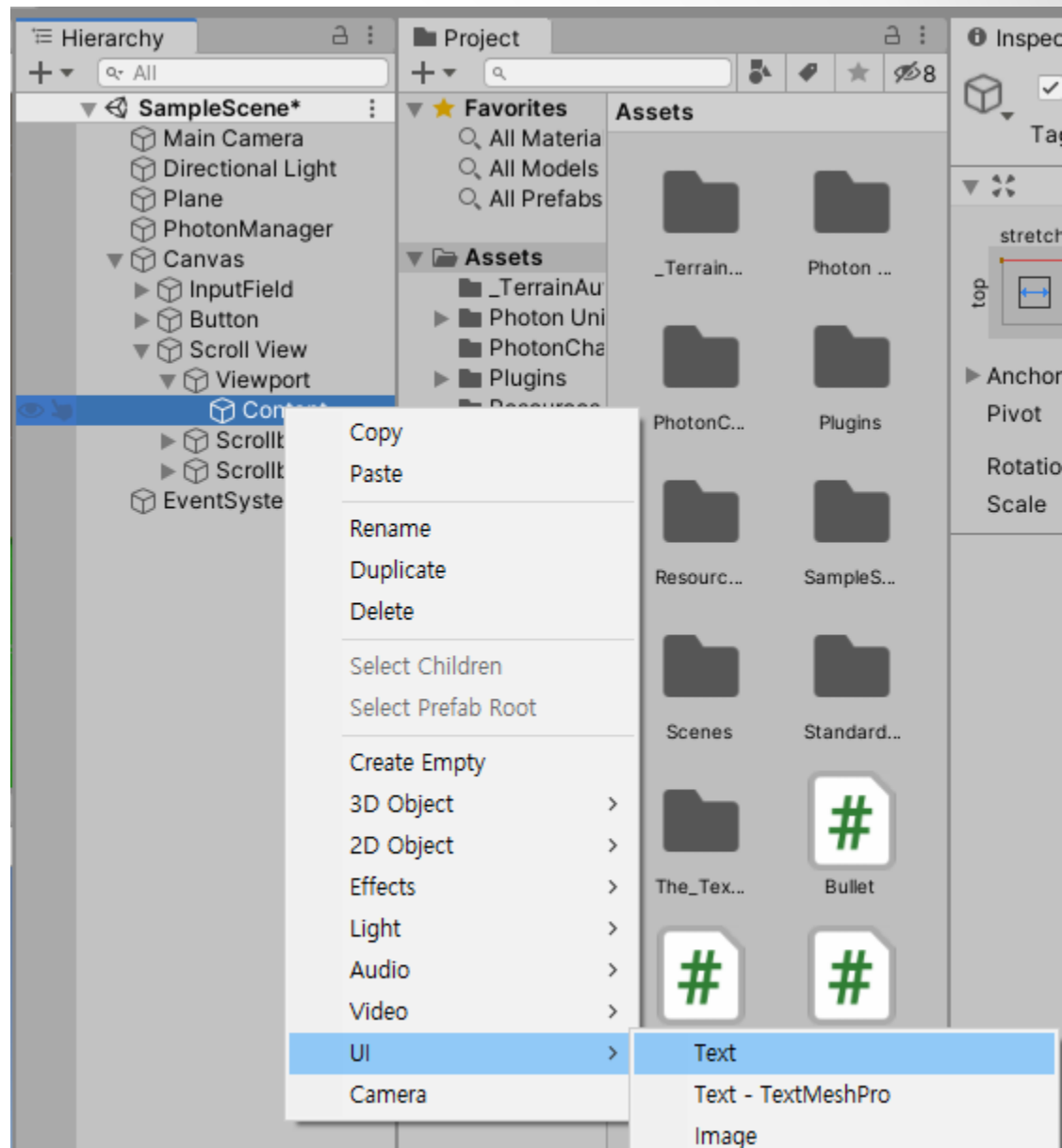
- Scroll View의 이름을 변경하고
- 위치를 다음과 같이 설정





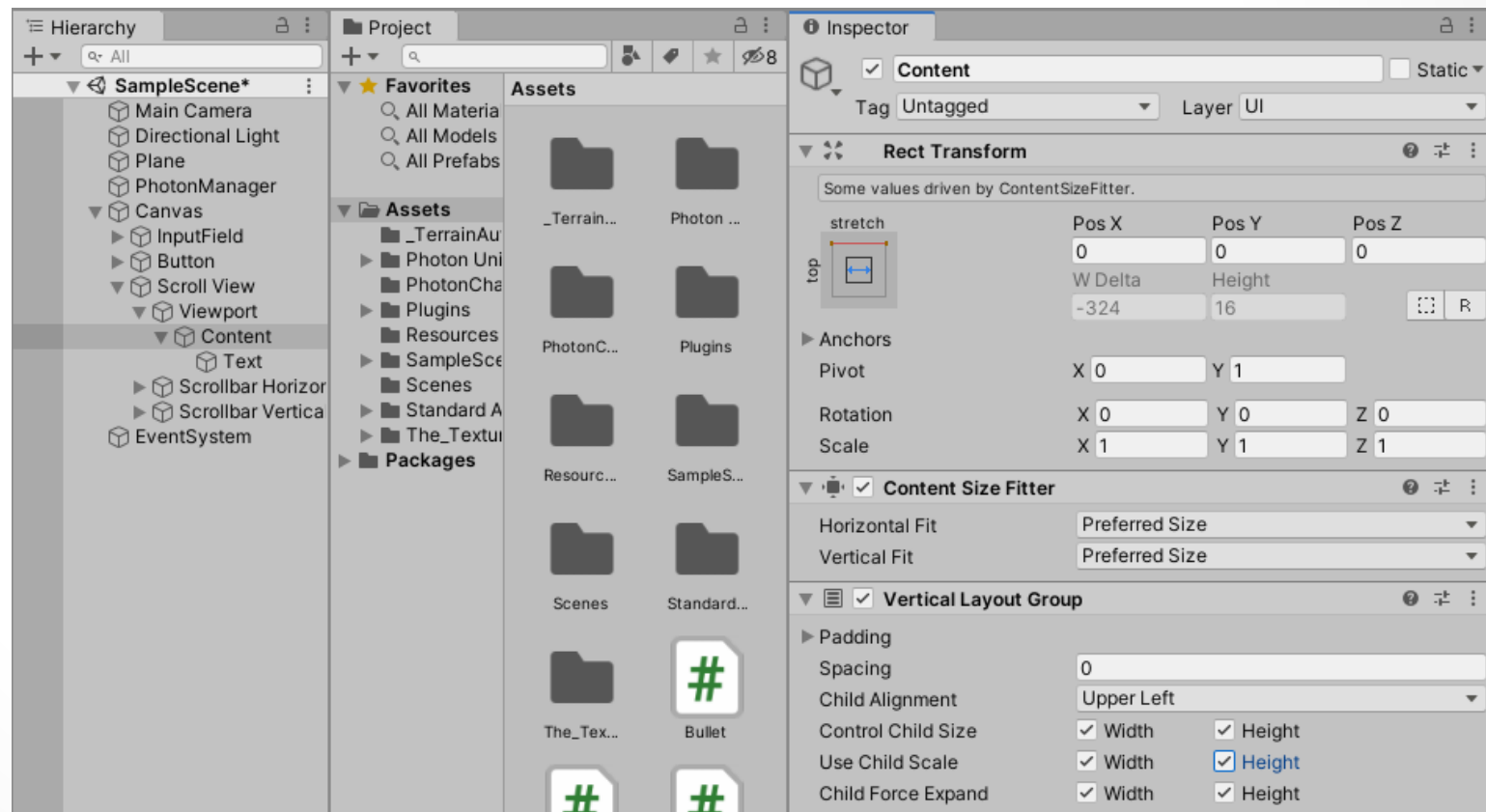
# 포톤네트워크 플레

- ScrollView-Viewport-Content 밑 Text 추가



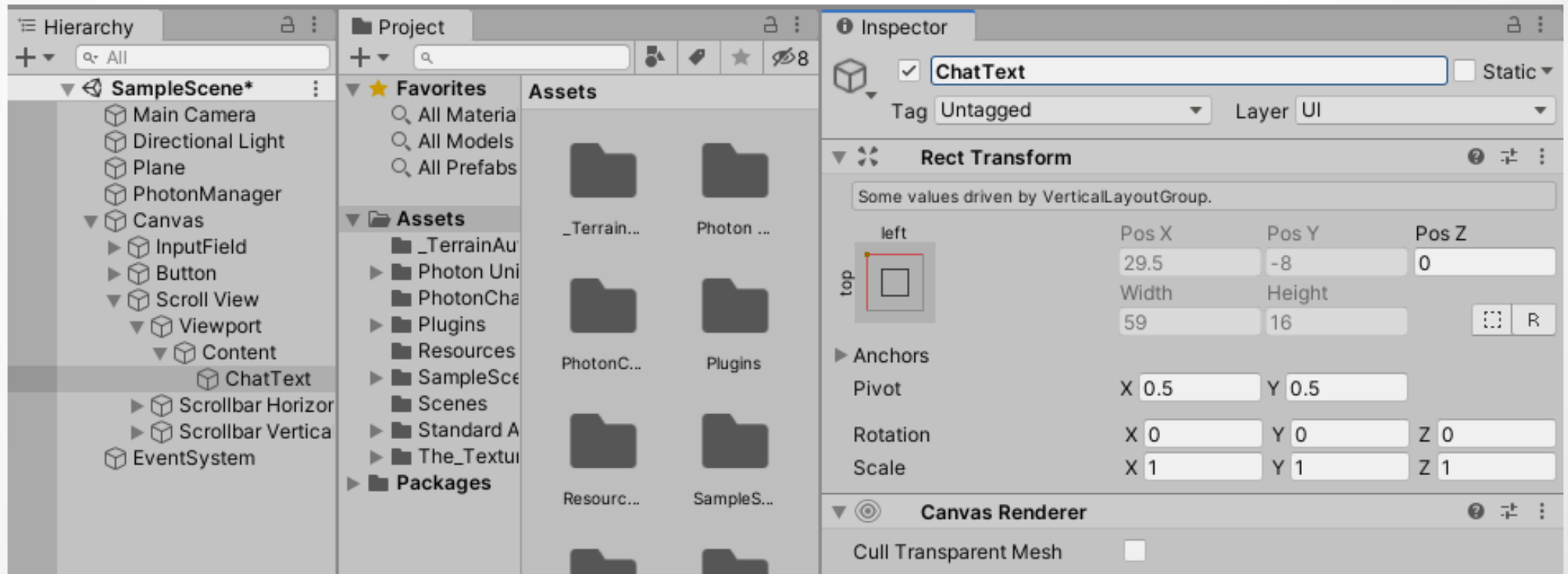
# 포톤네트워크 플레이어 정보 및 채팅

- Content 객체에서 Content Size Filter 및 Vertical Layout Group 추가



# 포톤네트워크 플레이어 정보 및 채팅

- Content 밑의 Text 이름을 ChatText로 변경



# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 파일에서
- chatMessage : 채팅용
- chatText
- Scroll\_rect

```
PhotonInit.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Photon;
5  using UnityEngine.UI;
6
7  public class PhotonInit : Photon.PunBehaviour
8  {
9      public InputField playerInput;
10     bool isGameStart = false;
11     string playerName = "";
12
13     public string chatMessage;
14     Text chatText;
15     ScrollRect scroll_rect = null;
16     void Awake()
17     {
18         PhotonNetwork.ConnectUsingSettings("MyFps 1.0");
19
20         chatText = GameObject.Find("ChatText").GetComponent<Text>();
21         scroll_rect = GameObject.Find("Scroll View").GetComponent<ScrollRect>();
22     }
```

# 포톤네트워크 플레이어 정보 및 채팅

- 다음의 함수를 추가해야함!

```
102 public void ShowChat(string chat)
103 {
104     // 게임 시작중에는 채팅 메시지로 처리
105     chatText.text += chat + "\n";
106     // 스크롤바의 위치를 제일 아래로 내려줌 1.0은 맨위, 0.0은 맨 아래
107     scroll_rect.verticalNormalizedPosition = 0.0f;
108 }
```

# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 파일에서 버튼클릭하는 SetPlayerName()을 수정
- 핵심은 isGameStart 변수로 게임 생성때 플레이어 이름을 적용하고, 그후에는 채팅 메시지로 활용함

```
83 public void SetPlayerName()
84 {
85     Debug.Log(playerInput.text + "를 입력 하셨습니다!");
86     if (isGameStart == false)
87     {
88         // 게임 시작을 안한경우 아이디 입력으로 처리
89         playerName = playerInput.text;
90         playerInput.text = string.Empty;
91         isGameStart = true;
92     }
93     else
94     {
95         chatMessage = playerInput.text;
96         playerInput.text = string.Empty;
97     } showChat(chatMessage);
98 }
```

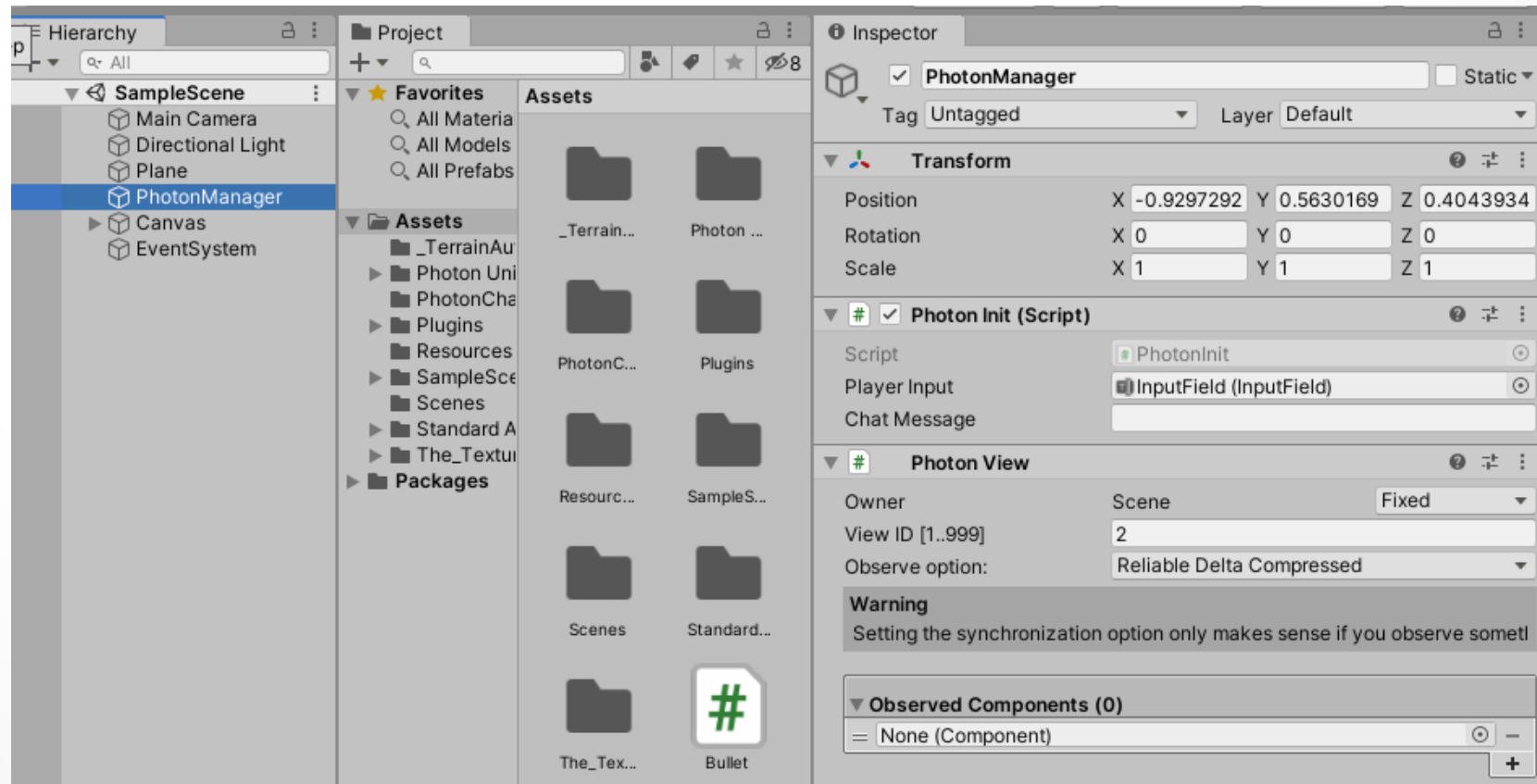
# 포톤네트워크 플레이어 정보 및 채팅

- 로컬 테스트 해보기
  - 텍스트 글자가 많아지면 스크롤바가 보이고 맨 밑으로 내려감



# 포톤네트워크 플레이어 정보 및 채팅

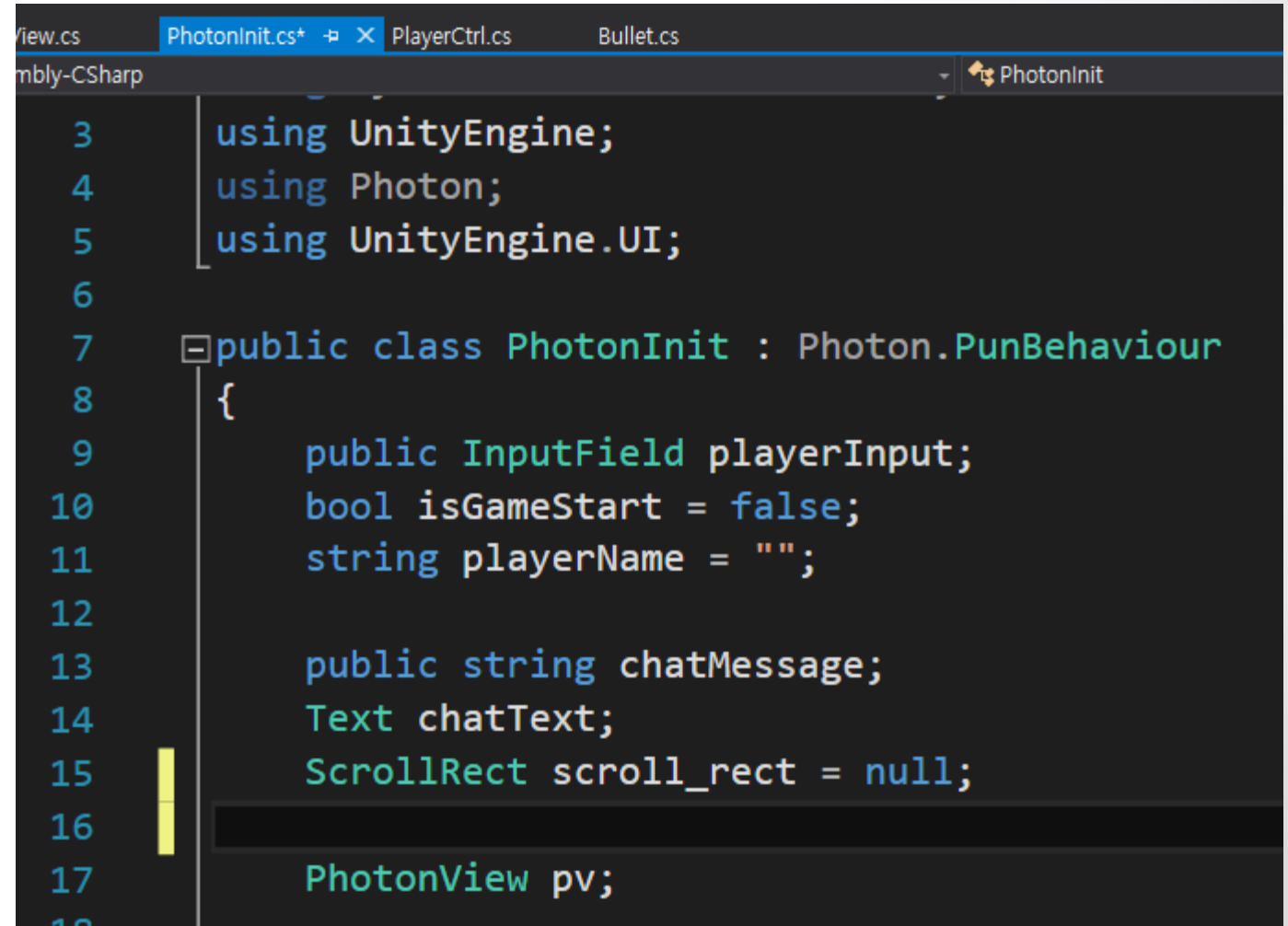
- 채팅 시스템을 포톤네트워크로 전달하기 위해선 PhotonManager 에 PhotonView를 추가해야함 (RPC 사용)





# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 에서 Photon View를 받을 변수 pv 추가



```
3 using UnityEngine;
4 using Photon;
5 using UnityEngine.UI;
6
7 public class PhotonInit : Photon.PunBehaviour
8 {
9     public InputField playerInput;
10    bool isGameStart = false;
11    string playerName = "";
12
13    public string chatMessage;
14    Text chatText;
15    ScrollRect scroll_rect = null;
16
17    PhotonView pv;
```

# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 에서 CreatePlayer에서 pv를 초기화해줘야함!
- 포톤뷰는 연결이 어느정도 이루어지고 나서 변수를 받아도 됨!

```
59  IEnumerator CreatePlayer()  
60  {  
61      while(!isGameStart)  
62      {  
63          yield return new WaitForSeconds(0.5f);  
64      }  
65  
66      GameObject tempPlayer = PhotonNetwork.Instantiate("Player",  
67                                              new Vector3(0, 1, 0),  
68                                              Quaternion.identity,  
69                                              0);  
70  
71      tempPlayer.GetComponent<PlayerCtrl>().SetPlayerName(playerName);  
72      pv = GetComponent<PhotonView>();  
73      yield return null;  
74  }
```

# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 에서 SetPlayerName()에서 RPC 콜을 날려야함!

```
82 public void SetPlayerName()
83 {
84     Debug.Log(playerInput.text + "를 입력 하셨습니다!");
85     if (isGameStart == false)
86     {
87         // 게임 시작을 안한경우 아이디 입력으로 처리
88         playerName = playerInput.text;
89         playerInput.text = string.Empty;
90         isGameStart = true;
91     }
92     else
93     {
94         chatMessage = playerInput.text;
95         pv.RPC("ChatInfo", PhotonTargets.All, chatMessage);
96         playerInput.text = string.Empty;
97     }
98 }
```

# 포톤네트워크 플레이어 정보 및 채팅

- PhotonInit.cs 에서 ChatInfo 추가 [PunRPC] 꼭 표시해야함!

```
110     [PunRPC]
111     public void ChatInfo(string sChat, PhotonMessageInfo info)
112     {
113         ShowChat(sChat);
114     }
```

```
102     public void ShowChat(string chat)
103     {
104         // 게임 시작중에는 채팅 메시지로 처리
105         chatText.text += chat + "\n";
106         // 스크롤바의 위치를 제일 아래로 내려줌 1.0은 맨위, 0.0은 맨 아래
107         scroll_rect.verticalNormalizedPosition = 0.0f;
108     }
```