

# | Unity Backend

“길드” 기능을 이용한 길드 생성, 가입

Created in 2023-07-05

Last Updated 2023-07-24

Unity Version 2022.2.2f1

## *Index*

- ◆ 길드 생성
- ◆ 에러 로그
- ◆ 길드 가입

# 길드 생성

- 길드 생성, 가입 조건 설정
- 길드 생성
- 길드 생성 페이지 제작



# 길드 생성

## ■ 길드 생성, 가입 조건 설정

- Backend Console에서 길드 생성, 가입 조건 설정

Backend Console << ProjectA ⚙️

ProjectA ⚠️ 개발 모드

★ 길드 관리 SDK 문서 > 콘솔 가이드 >

길드 **설정**

\*설정된 조건 값 이상에 해당하는 유저만 길드를 생성 또는 가입할 수 있습니다.

생성 조건 + 추가

가입 조건 선택해 주세요. 선택해 주세요. ↻

저장

길드 관리





# 길드 생성

## ■ Backend Console에서 길드 생성, 가입 조건 설정 (계속)

Backend Console << ProjectA ⚙️

ProjectA **개발 모드**

★ 길드 관리 SDK 문서 > 콘솔 가이드 >

길드 설정

\*설정된 조건 값 이상에 해당하는 유저만 길드를 생성 또는 가입할 수 있습니다.

생성 조건 **+** 추가

USER_DATA	level	2	🗑️
-----------	-------	---	----

가입 조건    ↻

**저장**

**길드를 생성하려면 레벨(level) 2 이상이 되어야 한다.**

Tip. 생성, 가입 조건은 "게임 정보 관리" 탭의 테이블 컬럼 값을 기준으로 설정

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침

© AFI, Inc. All rights reserved.



# 길드 생성

## ■ 길드 생성

### ■ 길드 생성 메소드

```
CreateGuildV3(string guildName, int goodsCount);
```

```
CreateGuildV3(string guildName, int goodsCount, CountryCode countryCode);
```

```
CreateGuildV3(string guildName, Param param, int goodsCount);
```

```
CreateGuildV3(string guildName, Param param, int goodsCount, CountryCode countryCode);
```

Value	설명	default
guildName	길드 이름	-
goodsCount	사용할 길드 재화의 개수 (1~10) (길드 생성 이후 수정 불가)	-
param	(Optional) 길드에 관한 메타정보	-
countryCode	(Optional) 국가 정보	CountryCode.NONE



# 길드 생성

## ■ 길드 생성 설명

뒤끝 콘솔에서 등록한 길드 생성 조건에 모두 만족하는 회원만 길드 생성 가능

- \* 길드명은 최소 2자 - 최대 16자이며, 한글, 영문, 숫자 사용 가능하며 특수문자는 불가능합니다.  
또한 숫자로 시작할 수 없으며, 공백은 가능하나 연속 입력 불가, 공백으로 시작하거나 끝날 수 없습니다.
- \* 길드 굿즈 종류의 개수는 생성시에만 정할 수 있으며 이후 종류의 개수를 줄이거나 늘리는 등의 변경은 불가능합니다.
- \* 메타정보는 원하는 정보를 저장할 수 있으며, ModifyGuild(Param)을 통해서 메타정보를 추가/수정 가능합니다.
- \* 국가 코드는 길드 생성 이후 길드 마스터가 UpdateCountryCode를 사용하여 추가/수정할 수 있습니다.
- \* 반드시 유저들의 국가 코드에 존재하는 국가만 길드의 국가 코드에 등록해야 합니다.  
유저들의 국가 코드 풀에 해당하지 않는 길드의 국가 코드는 콘솔에서 표시되지 않습니다.

### Error Cases

- \* goodsCount가 10 이상인 경우 (400 / bad goodsCount is too big, ..)
- \* goodsCount가 0 이하인 경우 (400 / undefined goodsCount must be more then 1, ..)
- \* 콘솔 설정 조건에 맞지 않는 유저가 길드 생성 시도한 경우 (403 / Forbidden createGuild, ..)
- \* 중복된 길드명으로 생성 시도한 경우 (409 / Duplicated guildName, ..)
- \* 길드명 조건이 맞지 않는 경우 (412 / guildName 사전 조건을 ..)
- \* 이미 속해 있는 길드가 존재하는 경우 (412 / JoinedGamer 사전 조건을 ..)



# 길드 생성

- 서버와 통신해 길드 관련 시스템을 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "BackendGuildSystem"으로 변경

```
1 using UnityEngine;
2 using BackEnd;
3
4 public class BackendGuildSystem : MonoBehaviour
5 {
6     public void CreateGuild(string guildName, int goodsCount=1)
7     {
8         Backend.Guild.CreateGuildV3(guildName, goodsCount, callback =>
9         {
10             if ( !callback.IsSuccess() )
11             {
12                 Debug.LogError($"길드 생성 도중 에러가 발생했습니다. : {callback}");
13
14                 return;
15             }
16
17             Debug.Log($"길드가 생성되었습니다. : {callback}");
18         });
19     }
20 }
```



# 길드 생성

- BackendSystem 오브젝트에 "BackendGuildSystem" 컴포넌트 추가

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the scene objects. Under the 'Lobby\*' folder, the 'BackendSystem' object is selected and highlighted with a red box. On the right, the Inspector panel shows the properties of the selected 'BackendSystem' component. The component list includes 'Transform', 'Backend Post System (Script)', 'Backend Coupon System (Script)', 'Backend Friend System (Script)', and 'Backend Guild System (Script)'. The 'Backend Guild System (Script)' component is highlighted with a red box, and its script 'BackendGuildSystem' is visible below it. An 'Add Component' button is located at the bottom of the Inspector panel.



# 길드 생성

- ButtonGameStart 오브젝트의 "RectTransform" 컴포넌트 변수 설정

The screenshot shows the Unity Inspector window with the **ButtonGameStart** object selected. The **Rect Transform** component is highlighted with a red dashed box. The settings for this component are as follows:

Property	Value
Anchor Presets	center
Pos X	0
Pos Y	400
Pos Z	0
Width	800
Height	200
Min X	0.5
Min Y	0
Max X	0.5
Max Y	0
Pivot X	0.5
Pivot Y	0
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1

Below the Rect Transform component, other components are visible: **Canvas Renderer**, **Image**, **Button**, and **Default UI Material (Material)**.



# 길드 생성

- 길드의 모든 것을 관리하는 길드 페이지 생성 및 설정
  - PopupBase 프리팹을 Hierarchy View로 Drag & Drop

**오브젝트의 이름을 PopupGuild로 변경**

**해당 팝업은 닫을 수 없기 때문에 Exit 오브젝트는 삭제합니다.**

Rect Transform	
stretch	Left: 50, Top: 600, Pos Z: 0
stretch	Right: 50, Bottom: 700
Anchors	
Min	X: 0, Y: 0
Max	X: 1, Y: 1
Pivot	X: 0.5, Y: 0.5
Rotation	
X	0
Y	0
Z	0
Scale	
X	1
Y	1
Z	1





# 길드 생성

- 길드의 모든 것을 관리하는 길드 페이지 생성 및 설정 (계속)

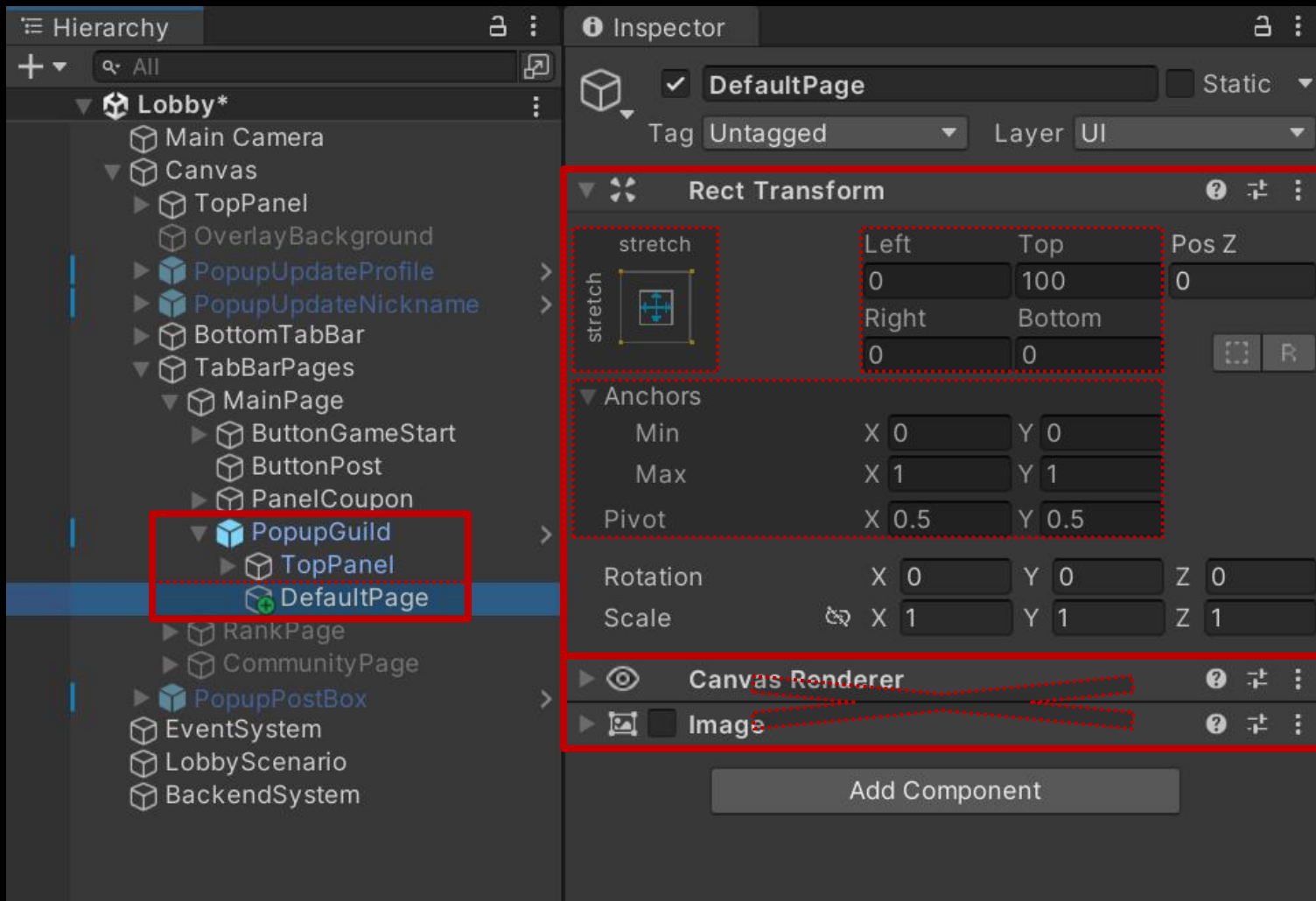
The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the scene's objects. The 'Lobby\*' folder is expanded, and the 'Title' object is selected under the 'PopupGuild' folder. On the right, the Inspector panel shows the properties for the selected 'TextMeshPro - Text (UI)' object. The 'Text Input' field contains the text '길드 (Guild)'. The 'Text Style' is set to 'Normal'. Under the 'Main Settings' section, the 'Font Asset' is set to 'NotoSansKR-Bold SDF (TMP\_Fc)', the 'Material Preset' is 'NotoSansKR-Bold SDF Material', and the 'Font Size' is 40. The 'Font Style' section shows buttons for Bold (B), Italic (I), Underline (U), Strikethrough (S), and other options. The 'Vertex Color' and 'Color Gradient' options are also visible.





# 길드 생성

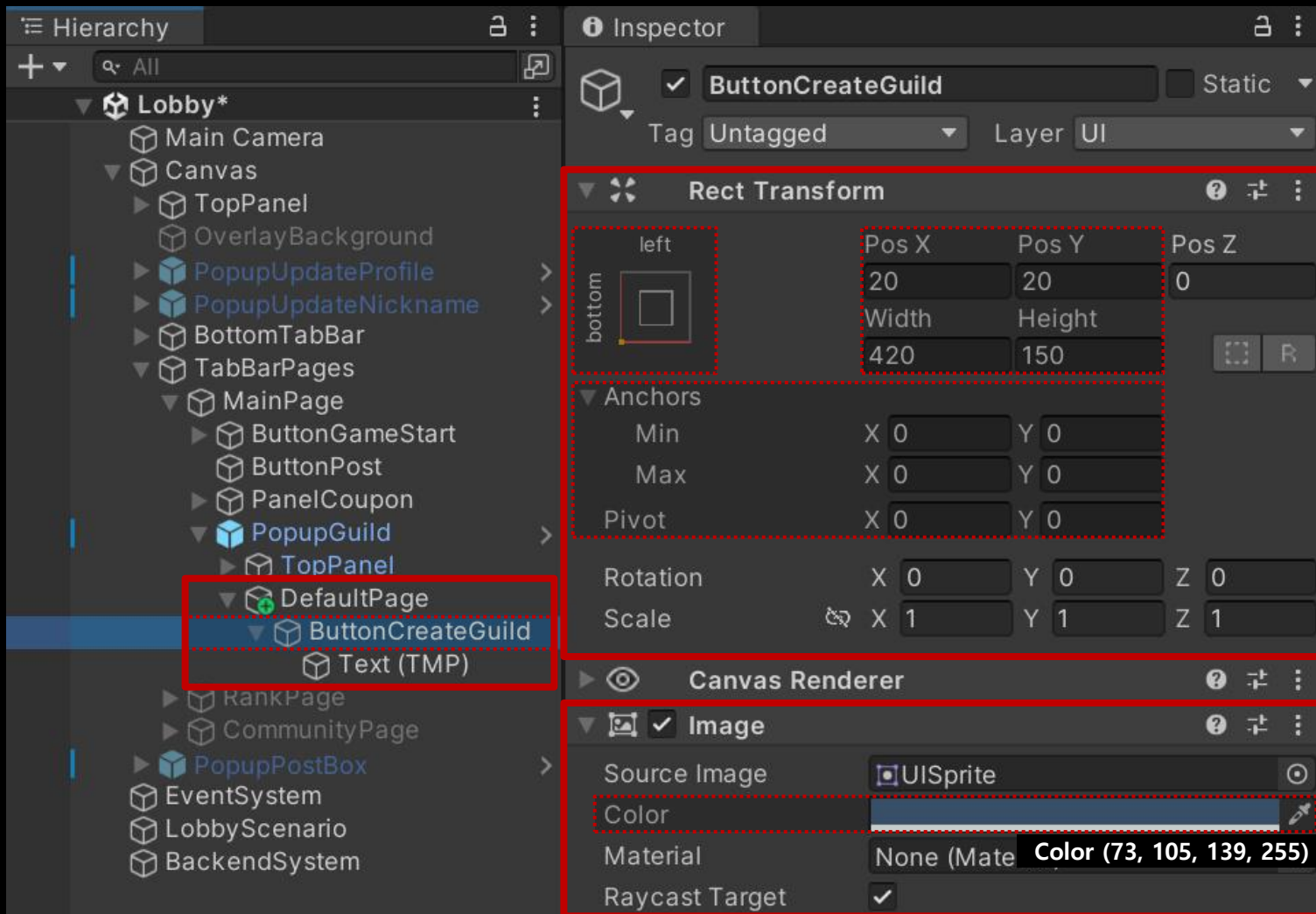
- 길드 기본 페이지의 UI를 관리하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel





# 길드 생성

- 길드 생성 페이지로 이동하는 "Button - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"





# 길드 생성

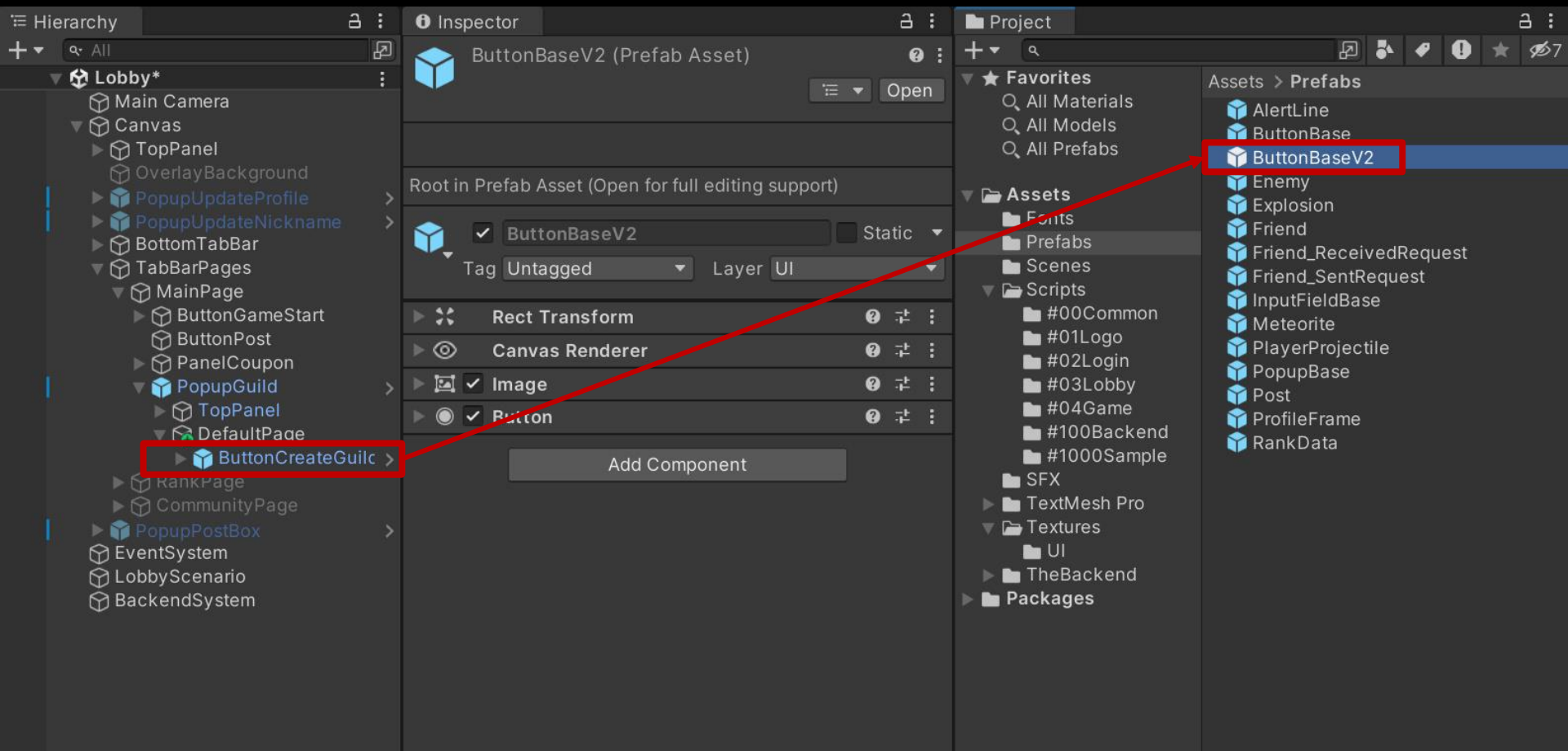
- 길드 생성 페이지로 이동하는 "Button - TextMeshPro" UI 생성 및 설정 (계속)

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree structure of the scene. The 'Lobby\*' folder is expanded, and the 'Text (TMP)' object under 'MainPage' is selected and highlighted with a red box. On the right, the Inspector panel shows the properties of the selected 'Text (TMP)' object. The 'TextMeshPro - Text (UI)' component is active and highlighted with a red box. The text content is '길드 생성'. The 'Font Asset' is set to 'NotoSansKR-Bold SDF (TMP\_...)'. The 'Font Style' is set to 'B' (Bold). The 'Font Size' is set to 50. The 'Vertex Color' is set to 'Color (255, 255, 255, 255)'. Other visible settings include 'Text Style' set to 'Normal', 'Material Preset' set to 'NotoSansKR-Bold SDF Material', and 'Font Style' buttons for 'B', 'I', 'U', 'S', 'ab', 'AB', and 'SC'.



# 길드 생성

- ButtonCreateGuild 오브젝트 Prefab 생성
  - Hierarchy View의 "ButtonCreateGuild" 오브젝트를 Project View로 Drag & Drop
  - 프리팹의 이름을 "ButtonBaseV2"로 변경





# 길드 생성

## ■ 길드 생성 페이지 제작

- 길드 생성 UI를 관리하는 길드 생성 페이지 생성 및 설정
  - PopupBase 프리팹을 Hierarchy View로 Drag & Drop

The screenshot displays the Unity development environment with three main panels:

- Hierarchy Panel:** Shows the scene hierarchy. A text box indicates: "오브젝트의 이름을 PopupCreateGuildPage로 변경" (Change the object's name to PopupCreateGuildPage). The selected object is "PopupCreateGuildPage", which is a child of "PopupUpdateNickname".
- Inspector Panel:** Shows the properties of the selected "PopupCreateGuildPage" object. The Prefab is set to "PopupBase". The "Rect Transform" component is highlighted with a red dashed box, showing its position and anchors. The "Canvas Renderer" and "Image" components are also visible.
- Project Panel:** Shows the Project window. The "Assets > Prefabs" folder is expanded, and "PopupBase" is highlighted with a red box.





# 길드 생성

- 길드 생성 UI를 관리하는 길드 생성 페이지 생성 및 설정 (계속)

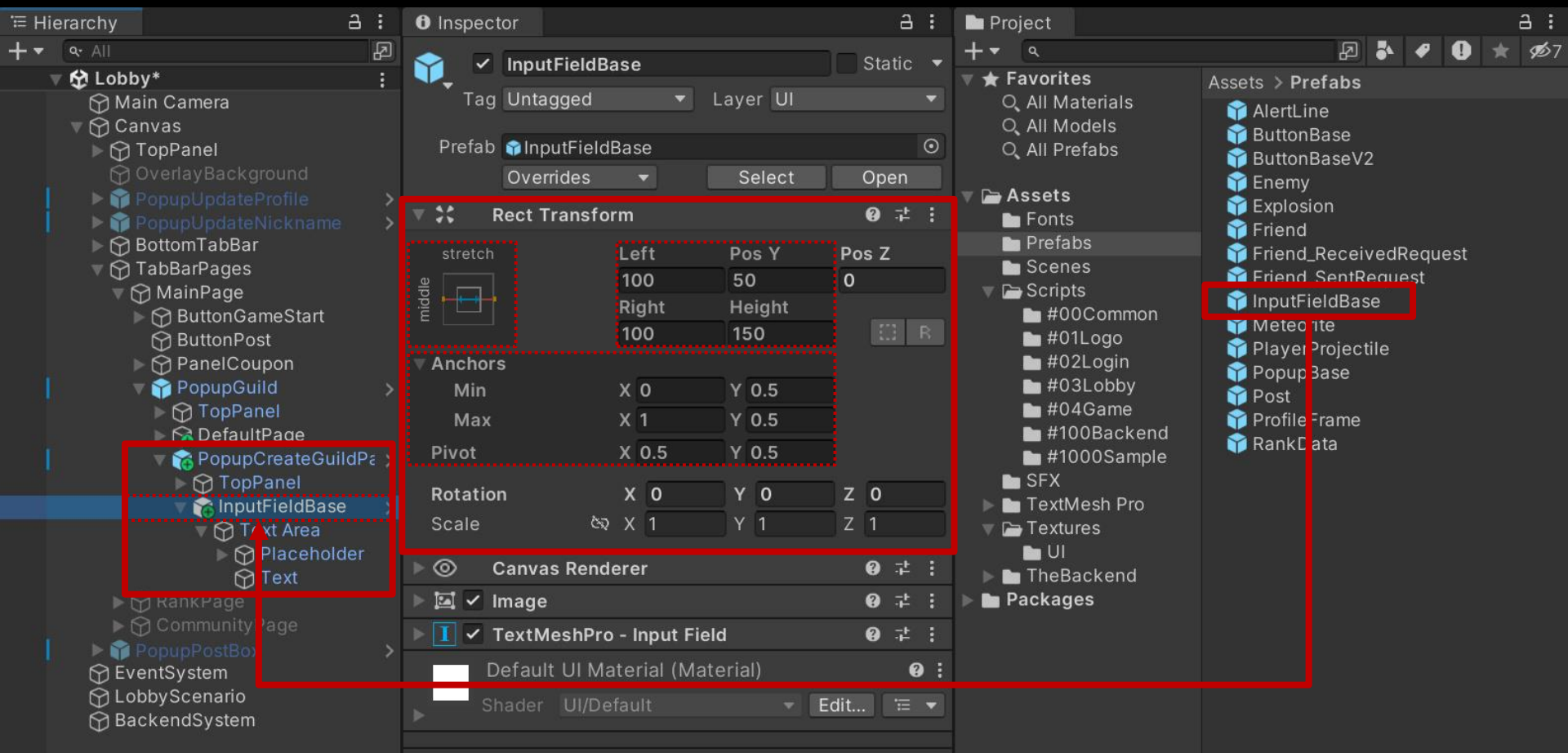
The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree structure under 'Lobby\*'. The 'Title' object is selected and highlighted with a red box. On the right, the Inspector panel shows the properties for the selected 'TextMeshPro - Text (UI)' component. A red box highlights the text input field, which contains the Korean text '길드 생성'. Below the text input, the 'Main Settings' section is visible, showing the following configurations:

- Text Style: Normal
- Font Asset: NotoSansKR-Bold SDF (TMP\_Fc)
- Material Preset: NotoSansKR-Bold SDF Material
- Font Style: B I U S ab AB SC
- Font Size: 40
- Auto Size:
- Vertex Color:
- Color Gradient:



# 길드 생성

- 생성할 길드 이름 입력 필드 생성 및 설정
  - InputFieldBase 프리팹을 Hierarchy View로 Drag & Drop





# 길드 생성

## ■ 생성할 길드 이름 입력 필드 생성 및 설정

The screenshot shows the Unity development environment with two panels: Hierarchy and Inspector.

**Hierarchy Panel:** Displays a tree view of the game object hierarchy. Under the "Lobby\*" object, a "Placeholder" and "Text" object are highlighted with a red box.

**Inspector Panel:** Shows the properties for the selected "TextMeshPro - Text (UI)" object. A red box highlights the following settings:

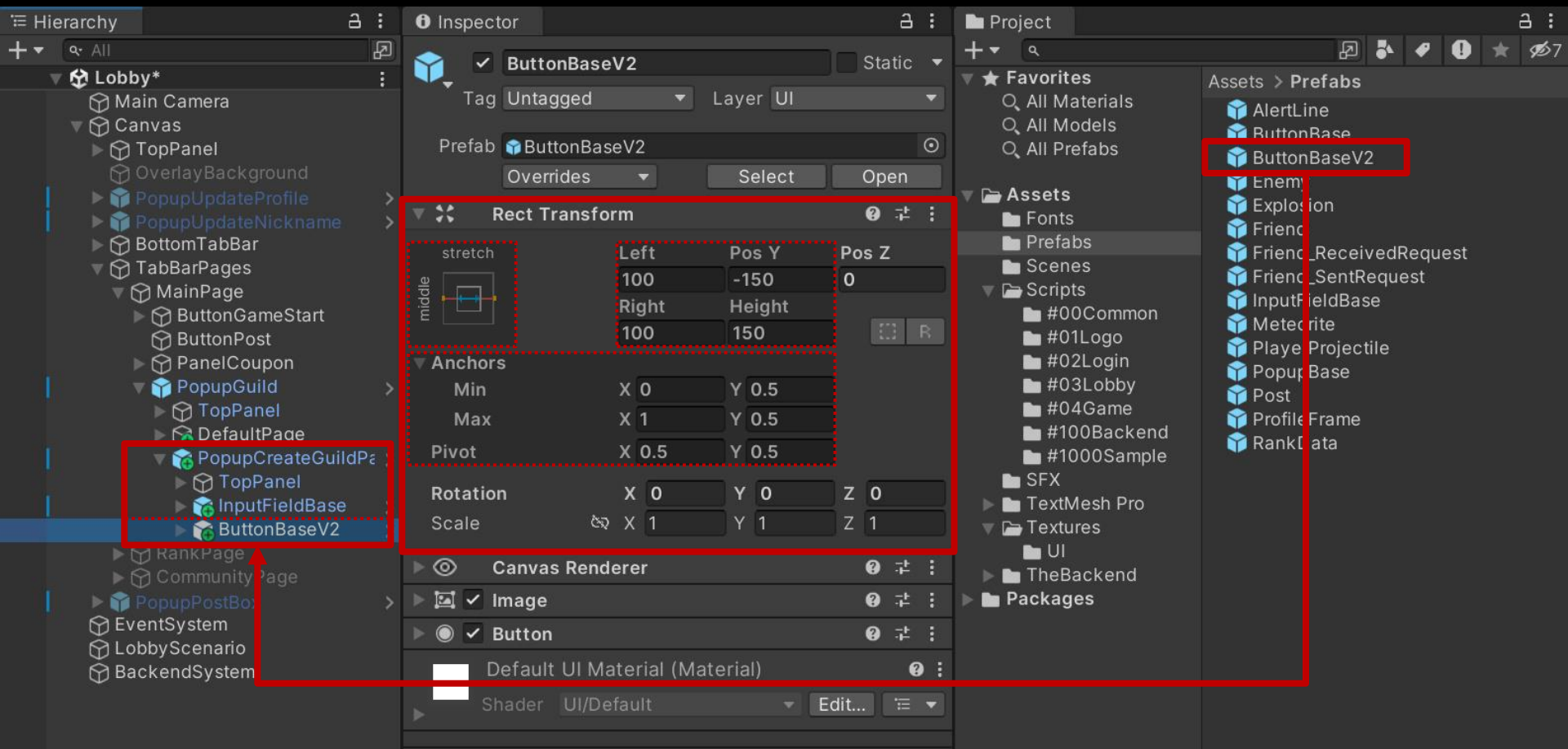
- Text Input:** The text field contains the placeholder text "Placeholder : 길드명을 입력하세요." (Placeholder: Enter guild name).
- Text Style:** Set to "Normal".
- Main Settings:**
  - Font Asset: NotoSansKR-Regular SDF (TMF)
  - Material Preset: NotoSansKR-Regular SDF Material
  - Font Style: B I U S ab AB SC
  - Font Size:** 60
  - Auto Size: [unchecked]
- Alignment:** The text is centered, indicated by the selected alignment icon (centered lines).





# 길드 생성

- 길드 생성 Button UI 생성 및 설정
  - ButtonBaseV2 프리팹을 Hierarchy View로 Drag & Drop





# 길드 생성

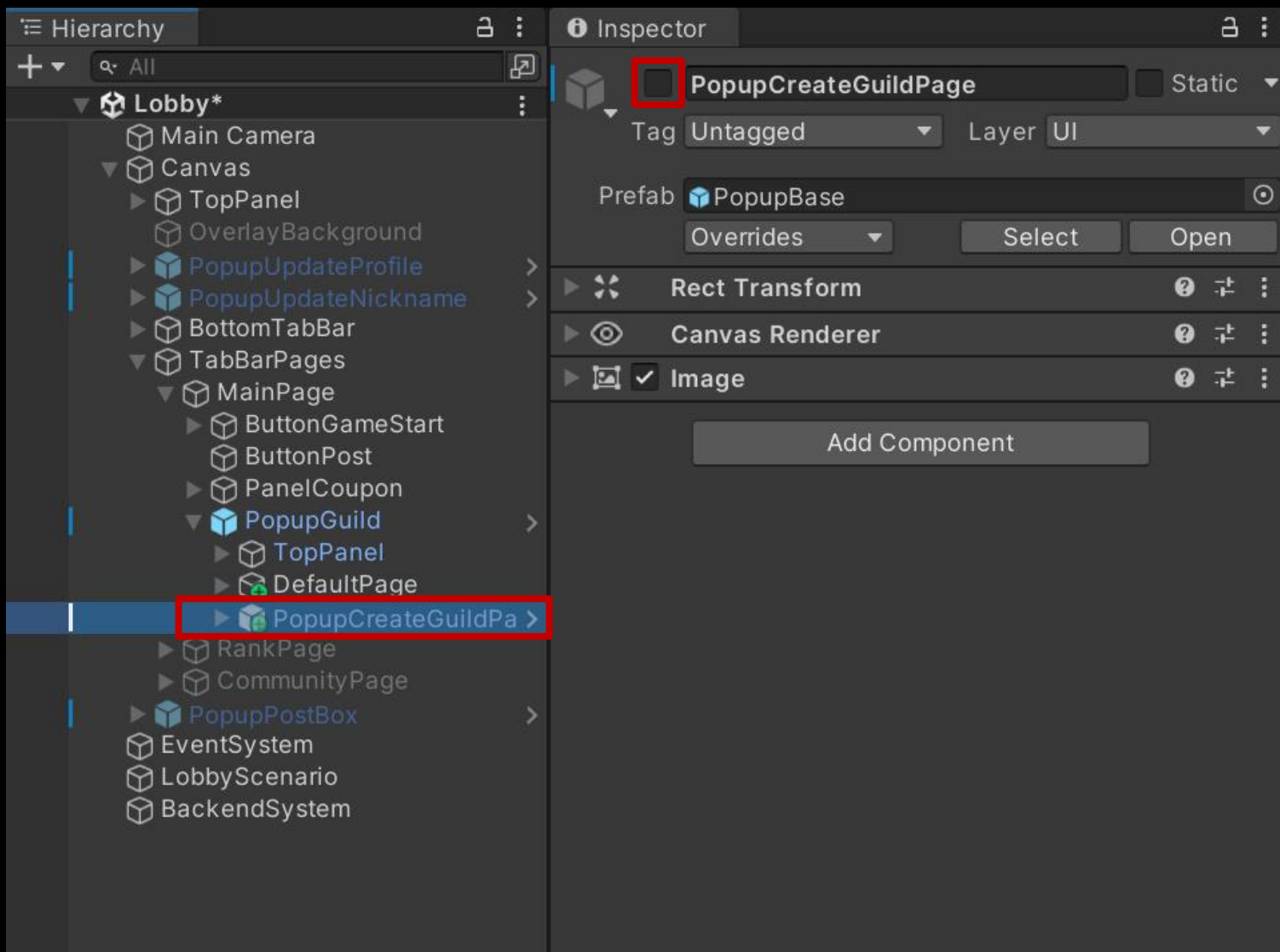
## ■ 길드 생성 Button UI 생성 및 설정 (계속)

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the 'Lobby\*' scene. The 'Text (TMP)' object is selected and highlighted with a red box. On the right, the Inspector panel shows the properties for the selected 'Text (TMP)' object. The 'TextMeshPro - Text (UI)' component is expanded, showing the 'Text Input' field with the text '길드 생성' and the 'Main Settings' section. The 'Main Settings' section includes 'Font Asset' (NotoSansKR-Bold SDF (TMP\_Fc)), 'Material Preset' (NotoSansKR-Bold SDF Material), 'Font Style' (B, I, U, S, ab, AB, SC), 'Font Size' (50), 'Auto Size' (unchecked), 'Vertex Color' (white), and 'Color Gradient' (unchecked).



# 길드 생성

- PopupCreateGuildPage 오브젝트 비활성화





# 길드 생성

- ButtonCreateGuild 오브젝트의 "Button" 컴포넌트 onClick 이벤트 설정

The screenshot displays the Unity Inspector for the **ButtonCreateGuild** object. The **Button** component is selected, and its **onClick ()** event is configured as follows:

- Runtime Only** (dropdown menu)
- GameObject.SetActive** (method dropdown)
- ButtonCreateGuild** (selected object)

Red dashed boxes highlight the **Runtime Only** dropdown, the **GameObject.SetActive** method, and the **ButtonCreateGuild** object selection. A red arrow points from the **ButtonCreateGuild** object in the Hierarchy panel to the selected object in the **onClick ()** event list.



# 길드 생성

- Exit 오브젝트의 "Button" 컴포넌트 onClick 이벤트 설정

The image shows the Unity development environment. On the left, the Hierarchy panel displays a tree view of the scene objects. The 'Exit' object is highlighted in blue. On the right, the Inspector panel shows the properties of the selected 'Exit' object. The 'Button' component is expanded, and the 'On Click ()' event is configured with the following settings:

- Transition: Color Tint
- Target Graphic: Exit (Image)
- Normal Color: [Color Picker]
- Highlighted Color: [Color Picker]
- Pressed Color: [Color Picker]
- Selected Color: [Color Picker]
- Disabled Color: [Color Picker]
- Color Multiplier: 1
- Fade Duration: 0.1
- Navigation: Automatic
- On Click (): Runtime Only, GameObject.SetActive, PopupCre





# 길드 생성

- 길드 생성 페이지를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildCreatePage"로 변경

```
1 using UnityEngine;
2 using TMPro;
3
4 public class GuildCreatePage : MonoBehaviour
5 {
6     [SerializeField]
7     private BackendGuildSystem backendGuildSystem;
8     [SerializeField]
9     private TMP_InputField inputFieldGuildName;
10
11     public void OnClickCreateGuild()
12     {
13         string guildName = inputFieldGuildName.text;
14
15         if ( guildName.Trim().Equals("") )
16         {
17             return;
18         }
19
20         inputFieldGuildName.text = "";
21
22         // 길드 생성
23         backendGuildSystem.CreateGuild(guildName);
24     }
25
26     public void SuccessCreateGuild()
27     {
28         gameObject.SetActive(false);
29     }
30 }
```



# 길드 생성

- PopupCreateGuildPage 오브젝트에 "GuildCreatePage" 컴포넌트 추가 및 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Lobby\*' object is expanded to show the 'PopupCreateGuildPage' component. The Inspector panel shows the configuration for the 'Guild Create Page (Script)' component. The 'Script' field is set to 'GuildCreatePage'. The 'Backend Guild System' field is set to 'BackendSystem (Backend Guild System)'. The 'Input Field Guild Name' field is set to 'InputFieldBase (TMP\_Input Field)'. Red boxes highlight these fields in the Inspector panel. Red arrows point from the 'BackendSystem' and 'InputFieldBase' fields in the Inspector panel to their respective components in the Hierarchy panel. The 'BackendSystem' component is also highlighted with a red box in the Hierarchy panel.



# 길드 생성

- ButtonBaseV2 오브젝트의 "Button" 컴포넌트 onClick 이벤트 설정

The screenshot displays the Unity Inspector for a **ButtonBaseV2** component. The **Button** sub-component is selected, and its **onClick ()** event is configured as follows:

- Runtime Only** (dropdown menu)
- GuildCreatePage.OnClickCreate** (selected event)

The **Target Graphic** is set to **ButtonBaseV2 (Image)**. The **Normal Color**, **Highlighted Color**, **Pressed Color**, and **Selected Color** are all set to white. The **Disabled Color** is set to a light gray. The **Color Multiplier** is set to 1, and the **Fade Duration** is set to 0.1. The **Navigation** is set to **Automatic**.





# 길드 생성

- 길드 생성에 성공했을 때 SuccessCreateGuild() 메소드 호출
  - BackendGuildSystem Script 수정

```
1  using UnityEngine;
2  using BackEnd;
3
4  public class BackendGuildSystem : MonoBehaviour
5  {
6      [SerializeField]
7      private GuildCreatePage guildCreatePage;
8
9      public void CreateGuild(string guildName, int goodsCount=1)
10     {
11         Backend.Guild.CreateGuildV3(guildName, goodsCount, callback =>
12         {
13             if ( !callback.IsSuccess() )...
14
15             Debug.Log($"길드가 생성되었습니다. : {callback}");
16
17             // 길드 생성에 성공했을 때 호출
18             guildCreatePage.SuccessCreateGuild();
19         });
20     }
21 }
22
23
24
25
26
```



# 길드 생성

- BackendSystem 오브젝트의 "BackendGuildSystem" 컴포넌트 변수 설정

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the scene objects. The 'BackendSystem' object is highlighted at the bottom. On the right, the Inspector panel shows the properties of the selected 'BackendSystem' object. The 'Backend Guild System (Script)' component is expanded, showing a variable named 'Guild Create Page' which is assigned to the 'PopupCreateGuildPage (Guild C...' script. A red box highlights the 'BackendSystem' object in the Hierarchy, the 'Backend Guild System (Script)' component in the Inspector, and the 'PopupCreateGuildPage' variable in both panels. A red arrow points from the variable in the Hierarchy to the variable in the Inspector.



# 길드 생성

## ■ 결과 화면

The screenshot displays a game development environment with a console window on the right. The console shows the following log messages:

```
UnityEditor.Log (object)
BackendGuildSystem/<>c:<CreateGuild>b_0_0 (BackEnd.BackendReturnO
#W.#V:#U ())

[12:20:54] 게임 정보 데이터 수정에 성공했습니다. : statusCode : 204
message : Success

[12:20:54] 데이터 조회에 성공했습니다 : statusCode : 200
message : Success

[12:20:55] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200
message : Success

[12:21:07] The character with Unicode value \uB2E1 was not found in t
UnityEngine.Debug:LogWarning (object,UnityEngine.Object)

[12:21:09] 길드가 생성되었습니다. : statusCode : 204
message : Success

길드가 생성되었습니다. : statusCode : 204
message : Success

UnityEngine.Debug:Log (object)
BackendGuildSystem/<>c:<CreateGuild>b_0_0 (BackEnd.BackendReturnO
#W.#V:#U ())
```

The game interface on the left shows a user profile for 'user03' with 30/30 hearts, 0 diamonds, and 1111 coins. Below the profile is a text input field labeled 'Enter text...' and a yellow button labeled '쿠폰등록'. A large grey box labeled '길드 (Guild)' is currently empty. At the bottom of the screen, there is a yellow button labeled '게임시작' and a row of five icons: a shop, a backpack, crossed swords, a tiered structure, and a person icon.



# 길드 생성

## ■ 결과 화면 (Backend Console)

Backend Console << ProjectA ⚙️ 🔊 📄 ? 6 관리자 👤

ProjectA ⚠️ 개발 모드

★ 길드 관리 🗑️ 삭제 전체 길드명으로 검색 SDK 문서 콘솔 가이드

길드 설정

<input type="checkbox"/>	번호	길드명	inDate	생성일	국가	길드마스터	닉네임	인원	굿즈	메타
<input type="checkbox"/>	1	유니티	2023-07-12T03:21:10.655Z	2023.07.12 12:21	-	d1d9de30-13ec-11ee-917d-c5adcf1b86b	user03	1	🔗	🔗
<input type="checkbox"/>	2	유니티노트	2023-07-11T07:15:13.803Z	2023.07.11 16:15	-	b9981f10-0859-11ee-9227-0f2b793d8148	닉네임	1	🔗	🔗

< 1 > 10개씩 보기

길드 관리

회사소개 이용약관 서비스수준협약 개인정보처리방침 © AFI, Inc. All rights reserved.



# 길드 생성

## ■ 결과 화면 (Backend Console)

Backend Console

ProjectA

길드 관리

유니티 길드원 목록

삭제

번호	유저 번호	닉네임	가입일	최종 접속일	등급
1	d1d9de30-13ec-11ee-917d-c5adcf1b86b	user03	2023.07.12 12:21	2023.07.12 12:20	길드마스터

1

10개씩 보기

확인

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침

© AFI, Inc. All rights reserved.

# 에러 로그

- 에러 로그 출력
- 에러 로그 UI 출력
- 에러 로그 서버 저장



# 에러 로그

## ■ 에러 로그 출력

### ■ 길드 생성에 실패했을 때 상황에 따라 에러 출력

#### □ BackendGuildSystem Script 수정

```
1  using UnityEngine;
2  using BackEnd;
3
4  public class BackendGuildSystem : MonoBehaviour
5  {
6      [SerializeField]
7      private GuildCreatePage guildCreatePage;
8
9      public void CreateGuild(string guildName, int goodsCount=1)
10     {
11         Backend.Guild.CreateGuildV3(guildName, goodsCount, callback =>
12         {
13             if ( !callback.IsSuccess() )
14             {
15                 //Debug.LogError($"길드 생성 도중 에러가 발생했습니다. : {callback}");
16                 ErrorLogCreateGuild(callback);
17
18                 return;
19             }
20
21             Debug.Log($"길드가 생성되었습니다. : {callback}");
22
23             // 길드 생성에 성공했을 때 호출
24             guildCreatePage.SuccessCreateGuild();
25         });
26     }
27 }
```





# 에러 로그

- BackendGuildSystem Script 수정 (계속)

```
28 private void ErrorLogCreateGuild(BackendReturnObject callback)
29 {
30     string message = string.Empty;
31
32     switch ( int.Parse(callback.GetStatusCode()) )
33     {
34         case 403: // Backend Console에 설정한 조건을 만족하지 못했을 때
35             message = "길드 생성을 위한 레벨이 부족합니다.";
36             break;
37         case 409: // 중복된 길드명으로 생성 시도한 경우
38             message = "이미 동일한 이름의 길드가 존재합니다.";
39             break;
40         default:
41             message = callback.GetMessage();
42             break;
43     }
44
45     // 에러 내용을 Console View에 출력
46     Debug.LogError($"길드 생성 도중 에러가 발생했습니다. : {message}");
47 }
48 }
```





# 에러 로그

## ■ 결과 화면

```
Console
Clear Collapse Error Pause Editor
[12:34:03] 길드 생성 도중 에러가 발생했습니다. : 이미 동일한 이름의 길드가 존재합니다.
UnityEngine.Debug:LogError (object)
[12:34:19] 길드 생성 도중 에러가 발생했습니다. : 길드 생성을 위한 레벨이 부족합니다.
UnityEngine.Debug:LogError (object)

길드 생성 도중 에러가 발생했습니다. : 길드 생성을 위한 레벨이 부족합니다.
UnityEngine.Debug:LogError (object)
BackendGuildSystem:ErrorLogCreateGuild (BackEnd.BackendReturnObject) (at Assets/Scripts/#100Backend/BackendGuildSystem.cs:40)
BackendGuildSystem:<CreateGuild>b_0_0 (BackEnd.BackendReturnObject) (at Assets/Scripts/#100Backend/BackendGuildSystem.cs:13)
#W.#V:#U ()
#W.#Z:#Y ()
```



# 에러 로그

## ■ 에러 로그 UI 출력

■ 에러 로그를 출력하는 "Text - TextMeshPro" UI 생성 및 설정

□ GameObject - UI - "Text - TextMeshPro"

The image shows a Unity development environment with three main panels:

- Hierarchy Panel:** Shows the scene structure. The **PopupGuild** object is selected, and its **TextLog** child is highlighted with a red box.
- Inspector Panel:** Shows the properties of the selected **TextLog** object. The **Rect Transform** component is expanded, with its **stretch** and **anchors** properties highlighted by red dashed boxes. The **Canvas Renderer** and **TextMeshPro - Text (UI)** components are also visible. The **Fade Effect\_TMP (Script)** component is expanded, with its **Effect Time** property highlighted by a red dashed box.
- TextMeshPro - Text (UI) Panel:** Shows the text input and styling options. The text **길드 로그 출력** is entered. The **Font Asset** is set to **NotoSansKR-Bold SDF (TMP\_Fc)**, and the **Font Size** is set to **40**. The **Alignment** is set to **Center**, and **Wrapping** is set to **Enabled**.



# 에러 로그

- 에러 내용을 FadeEffect\_TMP 타입의 UI에 출력
  - BackendGuildSystem Script 수정

```
1  using UnityEngine;
2  using BackEnd;
3
4  public class BackendGuildSystem : MonoBehaviour
5  {
6      [SerializeField]
7      private FadeEffect_TMP textLog;
8      [SerializeField]
9      private GuildCreatePage guildCreatePage;
10
11     public void CreateGuild(string guildName, int goodsCount=1) ...
12
13     private void ErrorLogCreateGuild(BackendReturnObject callback)
14     {
15         string message = string.Empty;
16
17         switch ( int.Parse(callback.GetStatusCode()) ) ...
18
19         // 에러 내용을 Console View에 출력
20         Debug.LogError($"길드 생성 도중 에러가 발생했습니다. : {message}");
21
22         // 에러 내용을 UI로 출력
23         textLog.FadeOut(message);
24     }
25 }
```



# 에러 로그

- BackendSystem 오브젝트의 "BackendGuildSystem" 컴포넌트 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'BackendSystem' object is selected under the 'Lobby\*' hierarchy. The Inspector panel shows the 'Backend Guild System (Script)' component with the following variables:

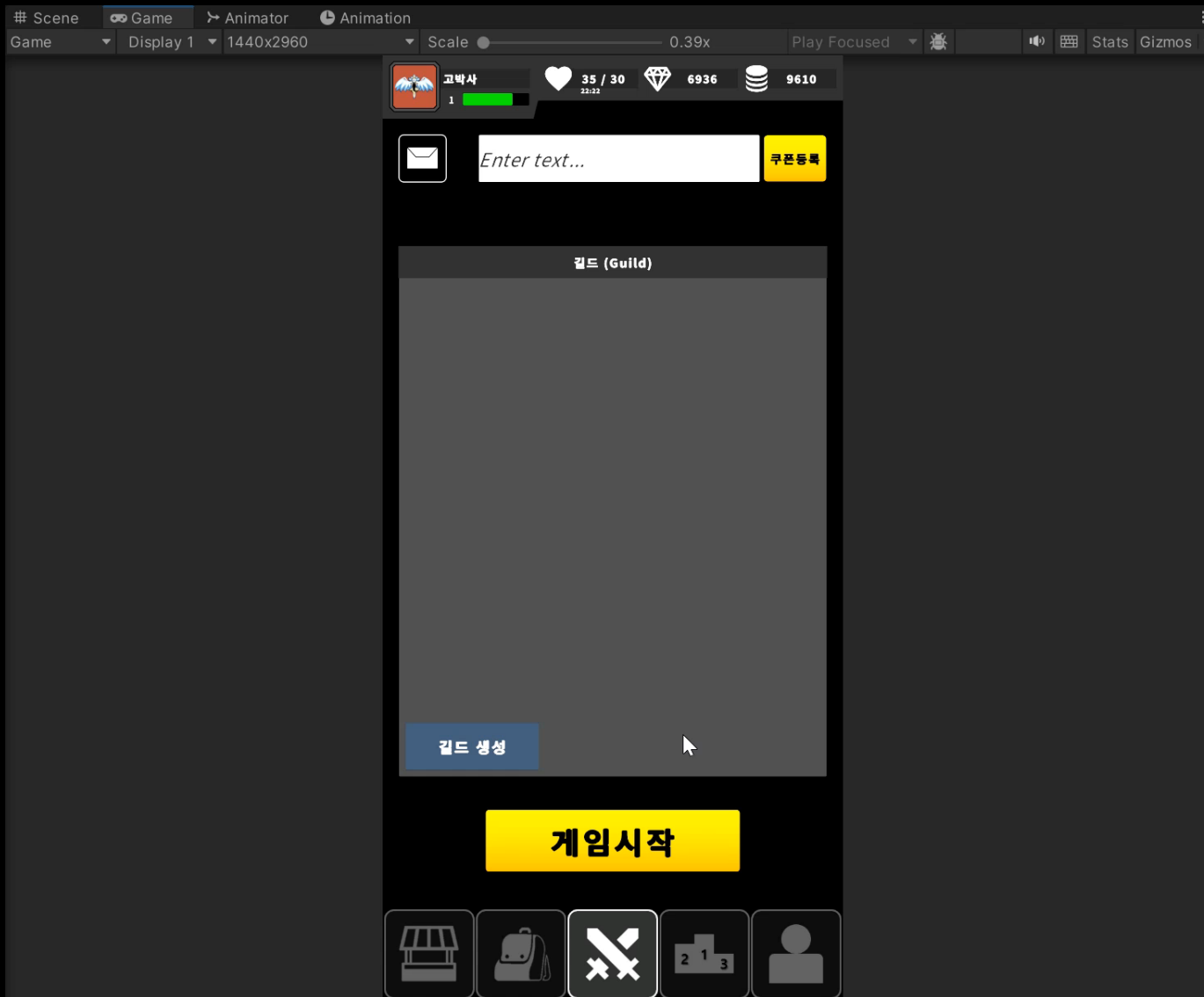
- Script: BackendGuildSystem
- Text Log: TextLog (Fade Effect\_TMP)
- Guild Create Page: PopupCreateGuildPage (Guild C...)

A red box highlights the 'Text Log' variable in the Inspector, and a red arrow points from it to the 'TextLog' object in the Hierarchy panel.



# 에러 로그

## ■ 결과 화면





# 에러 로그

## ■ 에러 로그 서버 저장

- InsertLog() 메소드를 호출해 에러 로그를 서버에 저장

- BackendGuildSystem Script 수정

```
1  +using ...
3
4  -public class BackendGuildSystem : MonoBehaviour
5  {
6      [SerializeField]
7      private FadeEffect_TMP textLog;
8      [SerializeField]
9      private GuildCreatePage guildCreatePage;
10
11  + public void CreateGuild(string guildName, int goodsCount=1) ...
28
29  - private void ErrorLogCreateGuild(BackendReturnObject callback)
30  {
31      string message = string.Empty;
32
33  + switch ( int.Parse(callback.GetStatusCode()) ) ...
45
46  - /*// 에러 내용을 Console View에 출력
47      Debug.LogError($"길드 생성 도중 에러가 발생했습니다. : {message}");
48
49      // 에러 내용을 UI로 출력
50      textLog.FadeOut(message);*/
51
52  + ErrorLog(message, "Guild_Failed_Log", "ApplyGuild");
53  }
54
```





# 에러 로그

- BackendGuildSystem Script 수정 (계속)

```
55 private void ErrorLog(string message, string behaviorType="", string paramKey="")
56 {
57     // 에러 내용을 Console View에 출력
58     Debug.LogError($"{paramKey} : {message}");
59
60     // 에러 내용을 UI로 출력
61     textLog.FadeOut(message);
62
63     // 에러 내용을 Backend Console에 저장
64     Param param = new Param() { { paramKey, message } };
65     // InsertLogV2(행동 유형, Key&Value)
66     Backend.GameLog.InsertLogV2(behaviorType, param);
67 }
68 }
```





# 에러 로그

## ■ 결과 화면

Backnd Console << ProjectA 🔧 🔊 📄 ? 6 관리자

ProjectA ⚠️ 개발 모드

★ 로그 관리 📄 CSV 다운로드

신규 로그 (InsertLogV2) 🔗 SDK 문서 🔗 콘솔 가이드 🔗

로그 관리 📄 퍼널 분석

행동 유형 📄 선택하세요

thebackend\_login\_success\_log

상세 검색 ▶

thebackend\_login\_fail\_log

Guild\_Failed\_Log

검색 🔍 검색 조건 초기화

▶ 번호 📄 유저 번호 📄 닉네임 📄 로그 생성 일시

로그가 없습니다.

📄 로그 관리

📄 영수증 검증

📄 길드 관리

📄 캐시 관리

📄 버전 관리

📄 공지사항

회사소개 📄 이용약관 📄 서비스수준협약 📄 개인정보처리방침

© AFI, Inc. All rights reserved.



# 에러 로그

## ■ 결과 화면

Backnd Console << ProjectA [설정] [알림] [도움말] [6] 관리자 [프로필]

ProjectA [경고] 개발 모드

- ★ 즐겨찾기
- 대시보드
- 서버 설정
- 뒤끝베이스
- 유저 관리
- 유저 접근 관리
- 게임 정보 관리
- 랭킹 관리
- 우편 관리
- 푸시 관리
- 쿠폰 관리
- 차트 관리
- % 확률 관리
- 로그 관리
- 영수증 검증
- 길드 관리
- 캐시 관리
- 버전 관리
- 공지사항

★ 로그 관리 CSV 다운로드

신규 로그(InsertLogV2) SDK 문서 콘솔 가이드

로그 관리 퍼널 분석

행동 유형 Guild\_Failed\_Log

상세 검색

검색 검색 조건 초기화

번호	유저 번호	닉네임	로그 생성 일시
1	aedaa3e0-0859-11ee-8b11-73576fc879f3	교박사	2023.07.12 12:46:40
<pre> {   "root": {     "items": [       {         "CREATEGUILD": "string \"길드 생성을 위한 ...\""       }     ]   } } </pre>			
2	b9981f10-0859-11ee-9227-0f2b793d8148	닉네임	2023.07.11 16:14:15
3	b9981f10-0859-11ee-9227-0f2b793d8148	닉네임	2023.07.11 16:14:10
4	aedaa3e0-0859-11ee-8b11-73576fc879f3	교박사	2023.07.10 22:32:52

< 1 > 10개씩 보기

회사소개 이용약관 서비스수준협약 개인정보처리방침 © AFI, Inc. All rights reserved.

# 길드 가입

- 길드 가입 요청
- 길드 가입 요청 유저 목록 조회
- 길드 가입 요청 승인
- 길드 가입 요청 거절



# 길드 가입

## ■ 길드 가입 요청

### ■ 길드 가입 요청 메소드

`ApplyGuildV3(string guildName);`

길드에 즉시 가입 설정이 ON으로 되어있을 경우 해당 길드에 즉시 가입된다.

길드에 즉시 가입 설정이 OFF로 되어있을 경우 길드 마스터 혹은 부 길드 마스터 승인 이후 길드에 가입된다.

#### Error Cases

- \* 콘솔 설정 조건에 맞지 않는 유저가 길드 가입 요청 시도한 경우 (403 / Forbidden applyGuild, ..)
- \* 이미 가입 요청한 길드에 다시 가입 요청 한 경우 (409 / Duplicated alreadyRequestGamer, ..)
- \* 이미 속해있는 길드가 존재하는 경우 (412 / JoinedGamer 사전 조건을 ..)
- \* 길드원이 이미 100명 이상인 경우 (429, guild member count 요청 횟수를 ..)



# 길드 가입

- 길드 이름에 해당하는 길드 InDate 반환 메소드

```
GetGuildInDateByGuildNameV3(string guildName);
```

매개변수 guildName 이름에 해당하는 길드의 inDate 조회

## Error Cases

\* 존재하지 않는 길드명일 경우 (404 / guild not found, ..)

\* 길드명을 입력하지 않은 경우 (404 / Resources not found, ..)

- 길드 JsonData

```
{  
  "guildInDate": "2023-06-26T02:22:22.022Z" // 길드의 inDate  
}
```



# 길드 가입

- ApplyGuild() 메소드를 호출해 길드 가입 요청
  - BackendGuildSystem Script 수정

```
1  using UnityEngine;
2  using BackEnd;
3  using System;
4
5  public class BackendGuildSystem : MonoBehaviour
6  {
7      [SerializeField]
8      private FadeEffect_TMP textLog;
9      [SerializeField]
10     private GuildCreatePage guildCreatePage;
11
12     public void CreateGuild(string guildName, int goodsCount=1)
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30     public void ApplyGuild(string guildName)
31     {
32         // GetGuildInDateByGuildNameV3() 메소드를 호출해 원하는 길드(guildName)의 guildInDate 정보 반환
33         string guildInDate = GetGuildInfoBy(guildName);
34
35         // guildInDate 정보를 가진 길드에 가입 요청을 보낸다.
36         Backend.Guild.ApplyGuildV3(guildInDate, callback =>
37         {
38             if ( !callback.IsSuccess() )
39             {
40                 ErrorLogApplyGuild(callback);
41
42                 return;
43             }
44
45             Debug.Log($"길드 가입 요청에 성공했습니다. : {callback}");
46         });
47     }
48 }
```



# 길드 가입

## BackendGuildSystem Script 수정 (계속)

```
49 public string GetGuildInfoBy(string guildName)
50 {
51     // 해당 길드명(guildName)의 길드가 존재하는지 여부는 동기로 진행
52     var bro = Backend.Guild.GetGuildInDateByGuildNameV3(guildName);
53     string inDate = string.Empty;
54
55     if ( !bro.IsSuccess() )
56     {
57         Debug.LogError($"길드 검색 도중 에러가 발생했습니다. : {bro}");
58         return inDate;
59     }
60
61     try
62     {
63         inDate = bro.GetFlattenJSON()["guildInDate"].ToString();
64
65         Debug.Log($"{{guildName}}의 inDate 값은 {{inDate}} 입니다.");
66     }
67     catch ( Exception e )
68     {
69         Debug.LogError(e);
70     }
71
72     return inDate;
73 }
74
```





# 길드 가입

## BackendGuildSystem Script 수정 (계속)

```
75 private void ErrorLogCreateGuild(BackendReturnObject callback)...
```

```
94
```

```
95 private void ErrorLogApplyGuild(BackendReturnObject callback)
```

```
96 {
```

```
97     string message = string.Empty;
```

```
98
```

```
99     switch ( int.Parse(callback.GetStatusCode()) )
```

```
100 {
```

```
101     case 403: // Backend Console에 설정한 조건을 만족하지 못했을 때
```

```
102         message = "길드 가입을 위한 레벨이 부족합니다.";
```

```
103         break;
```

```
104     case 409:
```

```
105         message = "이미 가입 요청한 길드입니다.";
```

```
106         break;
```

```
107     case 412:
```

```
108         message = "이미 다른 길드에 소속되어 있습니다.";
```

```
109         break;
```

```
110     case 429:
```

```
111         message = "길드에 더 이상 자리가 없습니다.";
```

```
112         break;
```

```
113     }
```

```
114
```

```
115     ErrorLog(message, "Guild_Failed_Log", "ApplyGuild");
```

```
116 }
```

```
117
```

```
118 private void ErrorLog(string message, string behaviorType="", string paramKey="")...
```

```
131 }
```



# 길드 가입

- 길드 검색 UI를 관리하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel

The screenshot displays the Unity Inspector window for a GameObject named 'SearchGuild'. The Hierarchy panel on the left shows the object's position within the 'DefaultPage' under the 'Lobby\*' scene. The Inspector panel shows the following components and settings:

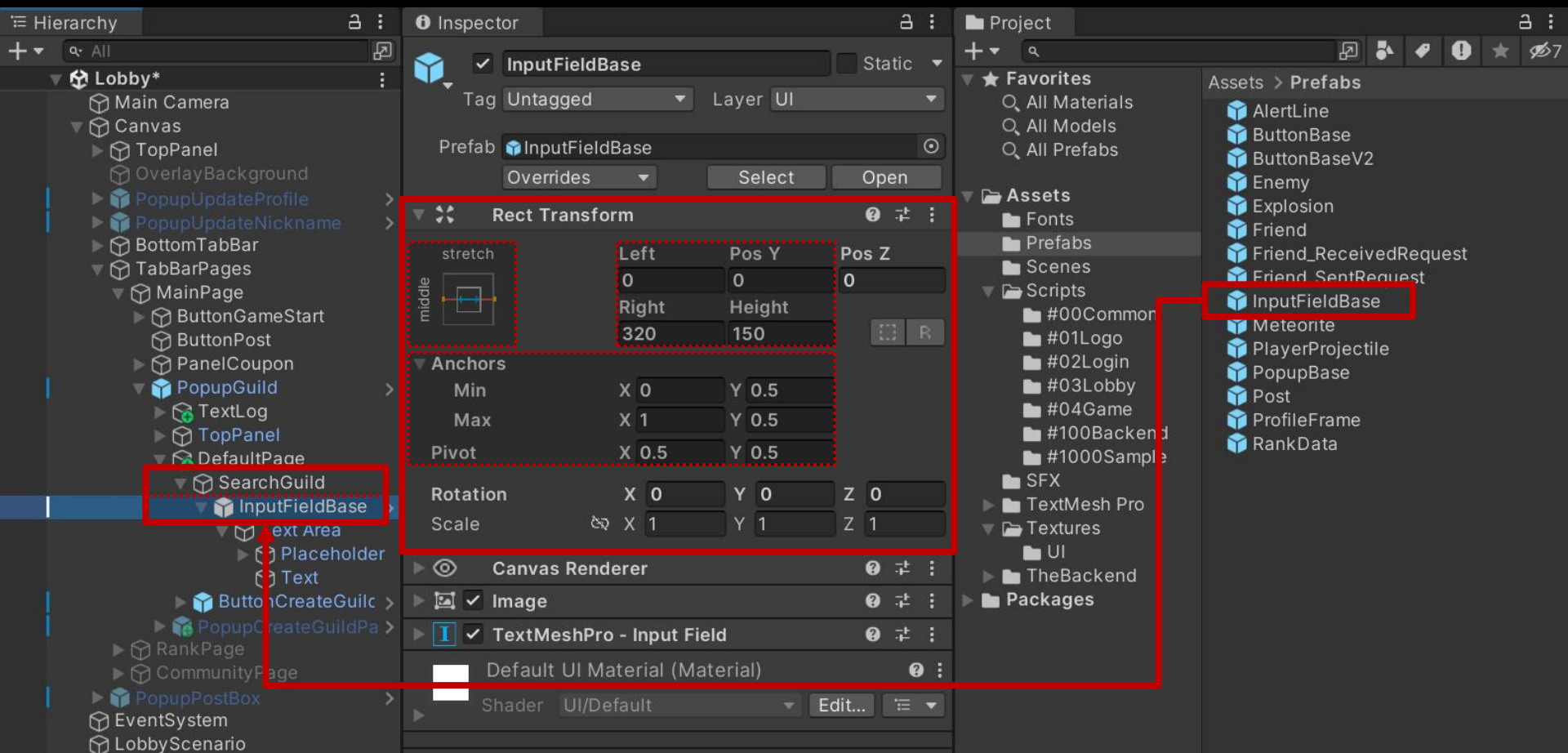
- SearchGuild** (UI): Tag: Untagged, Layer: UI.
- Rect Transform**:
  - stretch**: A red dashed box highlights the stretch icon and the 'top' anchor.
  - Left**: 20, **Pos Y**: -20, **Pos Z**: 0
  - Right**: 20, **Height**: 200
  - Anchors**:
    - Min**: X 0, Y 1
    - Max**: X 1, Y 1
    - Pivot**: X 0.5, Y 1
  - Rotation**: X 0, Y 0, Z 0
  - Scale**: X 1, Y 1, Z 1
- Canvas Renderer**: A red dashed box highlights the icon.
- Image**: A red dashed box highlights the icon.

An 'Add Component' button is visible at the bottom of the Inspector panel.



# 길드 가입

- 검색할 길드 이름 입력 필드 생성 및 설정
  - InputFieldBase 프리팹을 Hierarchy View로 Drag & Drop





# 길드 가입

- 검색할 길드 이름 입력 필드 생성 및 설정 (계속)

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the 'Lobby\*' scene. The 'SearchGuild' folder is expanded, and the 'Placeholder' object is selected. On the right, the Inspector panel shows the 'Rect Transform' component settings for the selected object. The 'stretch' property is set to 'stretch'. The 'anchors' section shows 'Min' at (0, 0) and 'Max' at (1, 1). The 'rotation' and 'scale' properties are also visible.

Property	X	Y	Z
Left	20	0	0
Right	0	0	0
Min	0	0	
Max	1	1	
Pivot	0.5	0.5	
Rotation	0	0	0
Scale	1	1	1





# 길드 가입

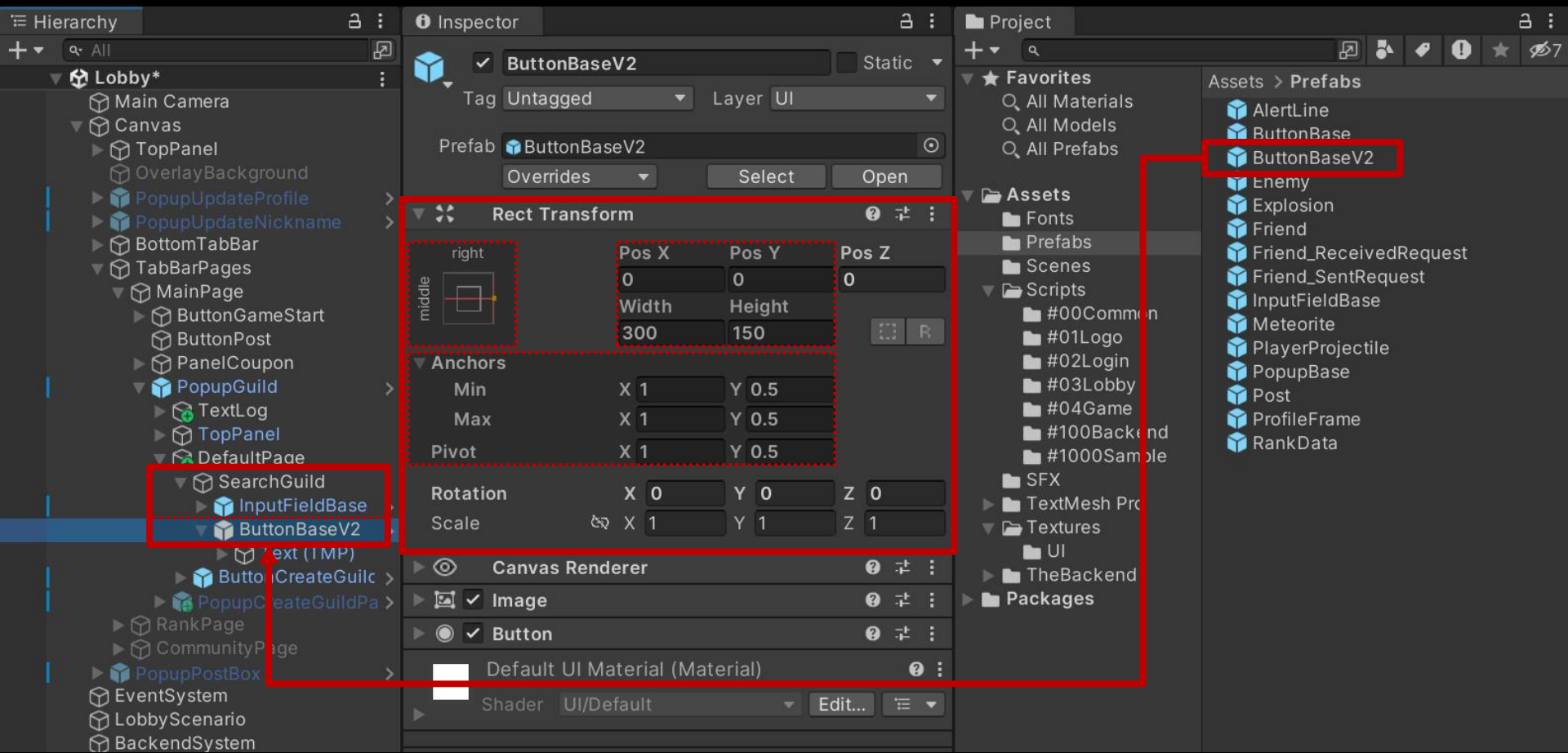
- 검색할 길드 이름 입력 필드 생성 및 설정 (계속)

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree view of the 'Lobby\*' scene. The 'Text Area' object is selected, and its children, 'Placeholder' and 'Text', are highlighted with a red box. On the right, the Inspector panel shows the properties for the selected 'TextMeshPro - Text (UI)' object. The 'Text' field contains the placeholder text 'Text : 검색할 길드명을 입력하세요.'. The 'Font Size' is set to 60, and the 'Alignment' is set to 'Left'. The 'Text Input' section is also visible, with the 'Enable RTL Editor' checkbox unchecked.



# 길드 가입

- 길드 검색 Button UI 생성 및 설정
  - ButtonBaseV2 프리팹을 Hierarchy View로 Drag & Drop





# 길드 가입

## ■ 길드 검색 Button UI 생성 및 설정 (계속)

The image shows the Unity development environment. On the left is the Hierarchy panel, and on the right is the Inspector panel.

**Hierarchy Panel:** Shows a tree view of the scene. The 'Lobby\*' folder is expanded, showing various UI elements. The 'Text (TMP)' object is selected under the 'SearchGuild' folder.

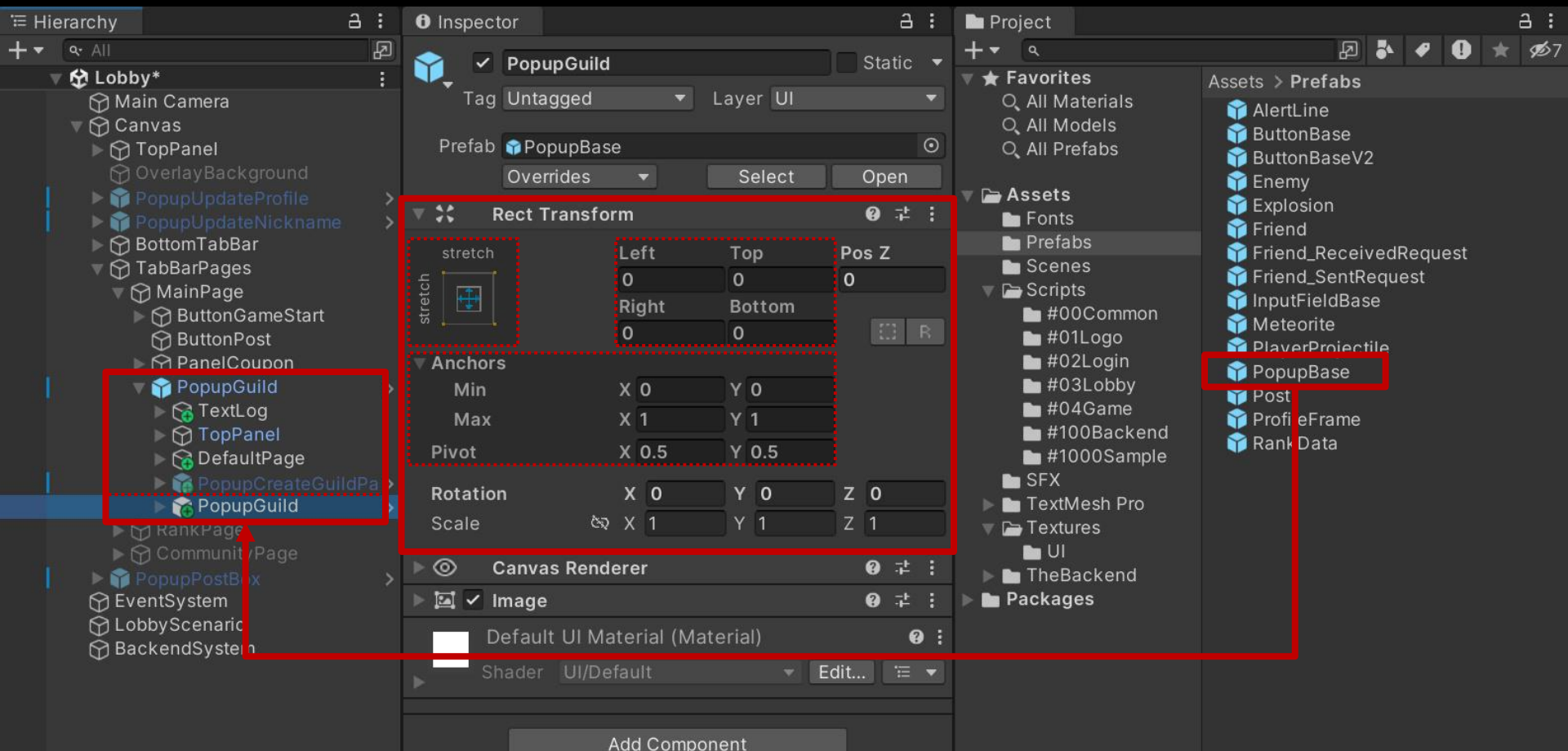
**Inspector Panel:** Shows the properties for the selected 'TextMeshPro - Text (UI)' object. The 'Text Input' field contains the text '길드 검색'. The 'Text Style' is set to 'Normal'. The 'Main Settings' section shows the 'Font Asset' is 'NotoSansKR-Bold SDF (TMP\_Fc)' and the 'Font Size' is '50'.





# 길드 가입

- 길드 페이지를 관리하는 UI 생성 및 설정
  - PopupBase 프리팹을 Hierarchy View로 Drag & Drop





# 길드 가입

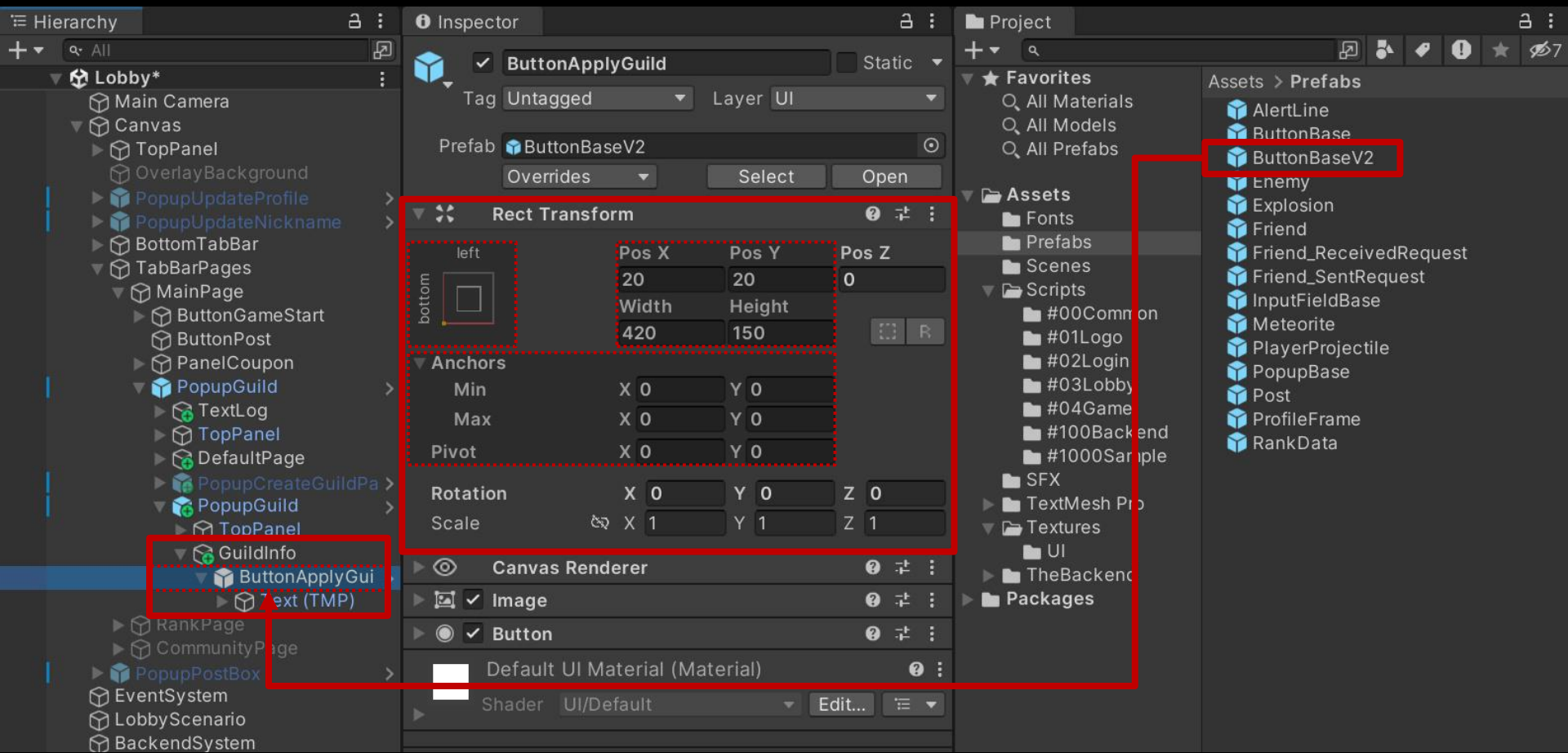
- 길드 정보를 출력하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree structure under 'Lobby\*'. The 'PopupGuild' object is selected, and its sub-objects 'TopPanel' and 'GuildInfo' are visible. A red dashed box highlights the 'GuildInfo' object in the Hierarchy. On the right, the Inspector panel shows the 'GuildInfo' component. The 'Rect Transform' component is highlighted with a red dashed box, showing the 'stretch' property for both width and height, and the 'Anchors' property set to 'Min X 0 Y 0' and 'Max X 1 Y 1'. The 'Canvas Renderer' component is also visible below it, also highlighted with a red dashed box. The 'Image' component is visible below the 'Canvas Renderer'.



# 길드 가입

- 길드 가입 요청 Button UI 생성 및 설정
  - ButtonBaseV2 프리팹을 Hierarchy View로 Drag & Drop





# 길드 가입

## ■ 길드 가입 요청 Button UI 생성 및 설정 (계속)

The screenshot shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left displays a tree structure under 'Lobby\*'. The 'MainPage' object is selected, and its child 'ButtonApplyGui' is expanded. The 'Text (TMP)' component under 'ButtonApplyGui' is highlighted with a red box. The Inspector panel on the right shows the properties for the selected 'Text (TMP)' component. The 'TextMeshPro - Text (UI)' component is selected, and its 'Text Input' field is highlighted with a red dashed box, containing the text '가입 신청'. Below the text input field, the 'Main Settings' are visible, including 'Font Asset' set to 'NotoSansKR-Bold SDF (TMP\_Fc...', 'Font Style' with 'B', 'I', 'U', 'S' buttons, 'Font Size' set to 50, and 'Vertex Color' set to white.



# 길드 가입

## ■ PopupGuild 오브젝트 비활성화

The screenshot displays the Unity interface with two panels: Hierarchy and Inspector.

**Hierarchy Panel:** Shows a tree structure under 'Lobby\*'. The 'PopupGuild' object under 'MainPage' is selected and highlighted with a red box.

**Inspector Panel:** Shows the properties for the selected 'PopupGuild' object. The 'Static' checkbox is checked and highlighted with a red box. The 'Image' component is visible and checked.

**Inspector Panel Details:**

- Static:
- Tag: Untagged
- Layer: UI
- Prefab: PopupBase
- Overrides: [Dropdown]
- Buttons: Select, Open
- Rect Transform: [Icon] [Help] [Reset] [Menu]
- Canvas Renderer: [Icon] [Help] [Reset] [Menu]
- Image:  [Icon] [Help] [Reset] [Menu]
- Button: Add Component





# 길드 가입

- Exit 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The image shows the Unity development environment. On the left is the Hierarchy panel, and on the right is the Inspector panel. The Hierarchy panel shows a tree structure under 'Lobby\*'. The 'Exit' object is selected and highlighted in blue. A red box highlights the 'Exit' object in the Hierarchy, and a red arrow points from it to the 'Button' component in the Inspector. The Inspector panel shows the 'Button' component settings, including 'Interactable', 'Transition', 'Normal Color', 'Highlighted Color', 'Pressed Color', 'Selected Color', 'Disabled Color', 'Color Multiplier', 'Fade Duration', 'Navigation', and 'On Click ()'. The 'On Click ()' event list contains two entries: 'GameObject.SetActive' and 'PopupGuil'. The 'Runtime Only' dropdown is also visible.



# 길드 가입

- 길드 페이지를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildPage"로 변경

```
1 using UnityEngine;
2 using TMPro;
3
4 public class GuildPage : MonoBehaviour
5 {
6     [SerializeField]
7     private BackendGuildSystem backendGuildSystem;
8     [SerializeField]
9     private TextMeshProUGUI textGuildName; // Popup 상단에 출력되는 길드 이름 Text UI
10
11     private string guildName = string.Empty; // 길드 이름
12
13     public void Activate(string guildName)
14     {
15         gameObject.SetActive(true);
16
17         textGuildName.text = guildName;
18         this.guildName = guildName;
19     }
20
21     public void OnClickApplyGuild()
22     {
23         backendGuildSystem.ApplyGuild(guildName);
24     }
25 }
```





# 길드 가입

- PopupGuild 오브젝트에 "PopupGuild" 컴포넌트 추가 및 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Lobby\*' folder is expanded to show the 'PopupGuild' component under the 'DefaultPage' folder. The 'BackendSystem' component is also visible at the bottom of the hierarchy. In the Inspector panel, the 'PopupGuild' component is selected, and its properties are shown. The 'Script' property is set to 'GuildPage'. The 'Backend Guild System' property is set to 'BackendSystem (Backend Guild)'. The 'Text Guild Name' property is set to 'Title (Text Mesh Pro UGUI)'. Red boxes highlight these three properties in the Inspector, and red arrows point from these boxes to the corresponding components in the Hierarchy panel.



# 길드 가입

- ButtonApplyGuild 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The screenshot displays the Unity Inspector window for a **ButtonApplyGuild** component. The **Button** sub-component is selected, and its **On Click ()** event list is visible. The event is configured with the following settings:

- Runtime Only**: A dropdown menu set to **Runtime Only**.
- Event Listener**: A dropdown menu set to **GuildPage.OnClickApplyGuild**.
- Target**: A button icon next to **PopupGuild**, which is highlighted with a red dashed box.

The **Inspector** also shows the **Button** component's properties, including **Interactable** (checked), **Transition** (Color Tint), and **Navigation** (Automatic). The **On Click ()** section includes a **Visualize** button and **+ -** controls.



# 길드 가입

- 길드 기본 페이지를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildDefaultPage"로 변경

```
1  using UnityEngine;
2  using TMPro;
3
4  public class GuildDefaultPage : MonoBehaviour
5  {
6      [SerializeField]
7      private BackendGuildSystem  backendGuildSystem;
8      [SerializeField]
9      private TMP_InputField      inputFieldGuildName;
10     [SerializeField]
11     private GuildPage            guildPage;
12     [SerializeField]
13     private FadeEffect_TMP       textLog;
14
```



# 길드 가입

- 길드 기본 페이지를 제어하는 스크립트 생성 및 작성 (계속)

```
15 public void OnClickSearchGuild()  
16 {  
17     string guildName = inputFieldGuildName.text;  
18  
19     if ( guildName.Trim().Equals("") )  
20     {  
21         return;  
22     }  
23  
24     inputFieldGuildName.text = "";  
25  
26     string inDate = backendGuildSystem.GetGuildInfoBy(guildName);  
27  
28     if ( inDate.Equals(string.Empty) )  
29     {  
30         textLog.FadeOut("존재하지 않는 길드입니다.");  
31     }  
32     else  
33     {  
34         // 길드 팝업 페이지 활성화  
35         guildPage.Activate(guildName);  
36     }  
37 }  
38 }
```



# 길드 가입

- DefaultPage 오브젝트에 "GuildDefaultPage" 컴포넌트 추가 및 설정

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the **DefaultPage** object selected. The Inspector panel on the right shows the properties of the selected object, including the **Guild Default Page (Script)** component. Red boxes and arrows highlight the configuration process:

- DefaultPage** is highlighted in the Hierarchy panel.
- BackendSystem** is highlighted in the Hierarchy panel, with an arrow pointing to its assignment in the Inspector.
- InputFieldBase** is highlighted in the Hierarchy panel, with an arrow pointing to its assignment in the Inspector.
- PopupGuild** is highlighted in the Hierarchy panel, with an arrow pointing to its assignment in the Inspector.
- TextLog** is highlighted in the Hierarchy panel, with an arrow pointing to its assignment in the Inspector.

The Inspector panel shows the following configuration for the **Guild Default Page (Script)** component:

- Script: GuildDefaultPage
- Backend Guild System: BackendSystem (Backend Guild System)
- Input Field Guild Name: InputFieldBase (TMP\_Input Field)
- Guild Page: PopupGuild (Guild Page)
- Text Log: TextLog (Fade Effect\_TMP)

The **Add Component** button is visible at the bottom of the Inspector panel.





# 길드 가입

- ButtonBaseV2 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The screenshot shows the Unity Inspector window with the following components and settings:

- Hierarchy:** Lobby\* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > MainPage > ButtonGameStart > ButtonPost > PanelCoupon > PopupGuild > TextLog > TopPanel > **DefaultPage** > SearchnGuild > InputFieldBase > **ButtonBaseV2** (selected)
- Inspector:**
  - ButtonBaseV2:** Tag: Untagged, Layer: UI, Prefab: ButtonBaseV2
  - Rect Transform**
  - Canvas Renderer**
  - Image**
  - Button:**
    - Interactable:
    - Transition: Color Tint
    - Target Graphic: ButtonBaseV2 (Image)
    - Normal Color: [Color Picker]
    - Highlighted Color: [Color Picker]
    - Pressed Color: [Color Picker]
    - Selected Color: [Color Picker]
    - Disabled Color: [Color Picker]
    - Color Multiplier: 1
    - Fade Duration: 0.1
    - Navigation: Automatic
  - On Click ():**
    - Runtime Only:
    - Event: GuildDefaultPage.OnClickSearch (selected)
    - Target: DefaultPa (selected)



# 길드 가입

## ■ 결과 화면

Unity Editor interface showing the game result screen and the console log.

**Game UI Elements:**

- Top bar: # Scene, Game, Animator, Animation, 0.39x Scale, Play Focused, Stats, Gizmos
- Player Info: 고박사 (1), 35 / 30 (22:22), 6936 (Diamonds), 9610 (Coins)
- Input: Enter text... (Coupon 등록)
- Guild (Guild) Section:
  - Input: 검색할 길드명을 입력하세요. (길드 검색)
  - Button: 길드 생성
- Start Button: 게임시작
- Bottom Bar: Shop, Backpack, Battle, Leaderboard, Profile

**Console Log (Success):**

```

[14:55:09] 길드 가입 요청에 성공했습니다. : statusCode : 204
message : Success

BackendGuildSystem:<ApplyGuild>b_3_0 (Backend.BackendReturnObject) (a
#W.#V:#U ()
  
```





# 길드 가입

## ■ 길드 가입 요청 유저 목록 조회

### ■ 길드 가입 요청 유저 목록 조회 메소드

```
GetApplicantsV3();  
GetApplicantsV3(int limit);  
GetApplicantsV3(int limit, int offset);
```

Value	설명	default
limit	가입 요청 유저 리스트 개수	100
offset	가입 요청 유저 리스트를 조회 할 시작점	0

#### Error Cases

\* 길드 마스터 혹은 운영진이 아닐 경우 (403 / Forbidden selectApplicant, ..)



# 길드 가입

## ■ 길드 가입 요청 유저 목록 JsonData

```
{
  "rows":
  [
    {
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "nickname": "고박사" // 유저의 닉네임
    },
    {
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "nickname": "이름없음" // 유저의 닉네임
    }
  ]
}
```



# 길드 가입

- 길드 멤버 데이터를 관리하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildMemberData"로 변경

```
1 public class GuildMemberData
2 {
3     public string nickname; // 닉네임
4     public string inDate; // 해당 유저의 inDate
5     public string level; // 해당 유저의 level
6
7     public override string ToString()
8     {
9         string result = string.Empty;
10        result += $"nickname : {nickname}\n";
11        result += $"inDate : {inDate}\n";
12        result += $"level : {level}\n";
13
14        return result;
15    }
16 }
```



# 길드 가입

- GetApplicants() 메소드를 호출해 길드 가입 요청 유저 목록 조회
  - BackendGuildSystem Script 수정

```
1  using System.Collections.Generic;
2  using UnityEngine;
3  using BackEnd;
4  using System;
5
6  public class BackendGuildSystem : MonoBehaviour
7  {
8      [SerializeField]
9      private FadeEffect_TMP      textLog;
10     [SerializeField]
11     private GuildCreatePage      guildCreatePage;
12
13     public void CreateGuild(string guildName, int goodsCount=1)...
30
31     public void ApplyGuild(string guildName)...
49
```



# 길드 가입

- BackendGuildSystem Script 수정 (계속)

```
50 public void GetApplicants()  
51 {  
52     Backend.Guild.GetApplicantsV3(callback =>  
53     {  
54         if ( !callback.IsSuccess() )  
55         {  
56             // 실패 사유가 403 하나 밖에 없기 때문에 별도로 메소드 제작 x  
57             ErrorLog(callback.GetMessage(), "Guild_Failed_Log", "GetApplicants");  
58  
59             return;  
60         }  
61  
62         // JSON 데이터 파싱 성공  
63         try  
64         {  
65             LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];  
66  
67             if ( jsonData.Count <= 0 )  
68             {  
69                 Debug.LogWarning("길드 가입 요청 목록이 비어있습니다.");  
70                 return;  
71             }  
72
```



# 길드 가입

## □ BackendGuildSystem Script 수정 (계속)

```
73 List<TransactionValue> transactionList = new List<TransactionValue>();
74 List<GuildMemberData> guildMemberDataList = new List<GuildMemberData>();
75
76 foreach ( LitJson.JsonData item in jsonData )
77 {
78     GuildMemberData guildMember = new GuildMemberData();
79
80     guildMember.nickname = item["nickname"].ToString().Equals("True") ? "NONAME" : item["nickname"].ToString();
81     guildMember.inDate = item["inDate"].ToString();
82
83     guildMemberDataList.Add(guildMember);
84
85     // guildMember.inDate를 가지는 친구의 UserGameData 정보 불러오기
86     Where where = new Where();
87     where.Equal("owner_inDate", guildMember.inDate);
88     transactionList.Add(TransactionValue.SetGet(Constants.USER_DATA_TABLE, where));
89 }
90
```





# 길드 가입

## BackendGuildSystem Script 수정 (계속)

```
91 Backend.GameData.TransactionReadV2(transactionList, callback =>
92 {
93     if ( !callback.IsSuccess() )
94     {
95         ErrorLog(callback.GetMessage(), "Guild_Failed_Log", "GetApplicants-TransactionReadV2");
96         return;
97     }
98
99     LitJson.JsonData userData = callback.GetFlattenJSON()["Responses"];
100
101     if ( userData.Count <= 0 )
102     {
103         Debug.LogWarning($"데이터가 존재하지 않습니다.");
104         return;
105     }
106
107     for ( int i = 0; i < userData.Count; ++ i )
108     {
109         guildMemberDataList[i].level = userData[i]["level"].ToString();
110         Debug.Log(guildMemberDataList[i].ToString());
111     }
112     });
113 }
114 // JSON 데이터 파싱 실패
115 catch ( Exception e )
116 {
117     // try-catch 에러 출력
118     Debug.LogError(e);
119 }
120 });
121 }
122
123 public string GetGuildInfoBy(string guildName)...
```



# 길드 가입

- 길드 가입 요청 유저 목록 버튼을 배치하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel

The screenshot shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby\*' with various UI elements. The 'ExecutivesOption' component is selected under 'MainPage' > 'PopupGuild' > 'GuildInfo'. The Inspector panel on the right shows the properties of the 'ExecutivesOption' component, which is currently set to 'Rect Transform'. The 'Rect Transform' component is highlighted with a red dashed box, showing its 'stretch' mode set to 'top' and its 'Anchors' set to 'Min X: 0, Y: 1' and 'Max X: 1, Y: 1'. The 'Canvas Renderer' and 'Image' components are also visible below it. A red box highlights the 'ExecutivesOption' component in the Hierarchy panel. A text box at the bottom right contains the instruction: '생성한 오브젝트의 이름을 ExecutivesOption으로 변경'.

**Inspector Panel Properties:**

- Component: ExecutivesOption
- Tag: Untagged
- Layer: UI
- Rect Transform:
  - stretch: top
  - Left: 0, Pos Y: 0, Pos Z: 0
  - Right: 0, Height: 120
  - Anchors: Min (X: 0, Y: 1), Max (X: 1, Y: 1), Pivot (X: 0.5, Y: 1)
  - Rotation: X: 0, Y: 0, Z: 0
  - Scale: X: 1, Y: 1, Z: 1
- Canvas Renderer
- Image

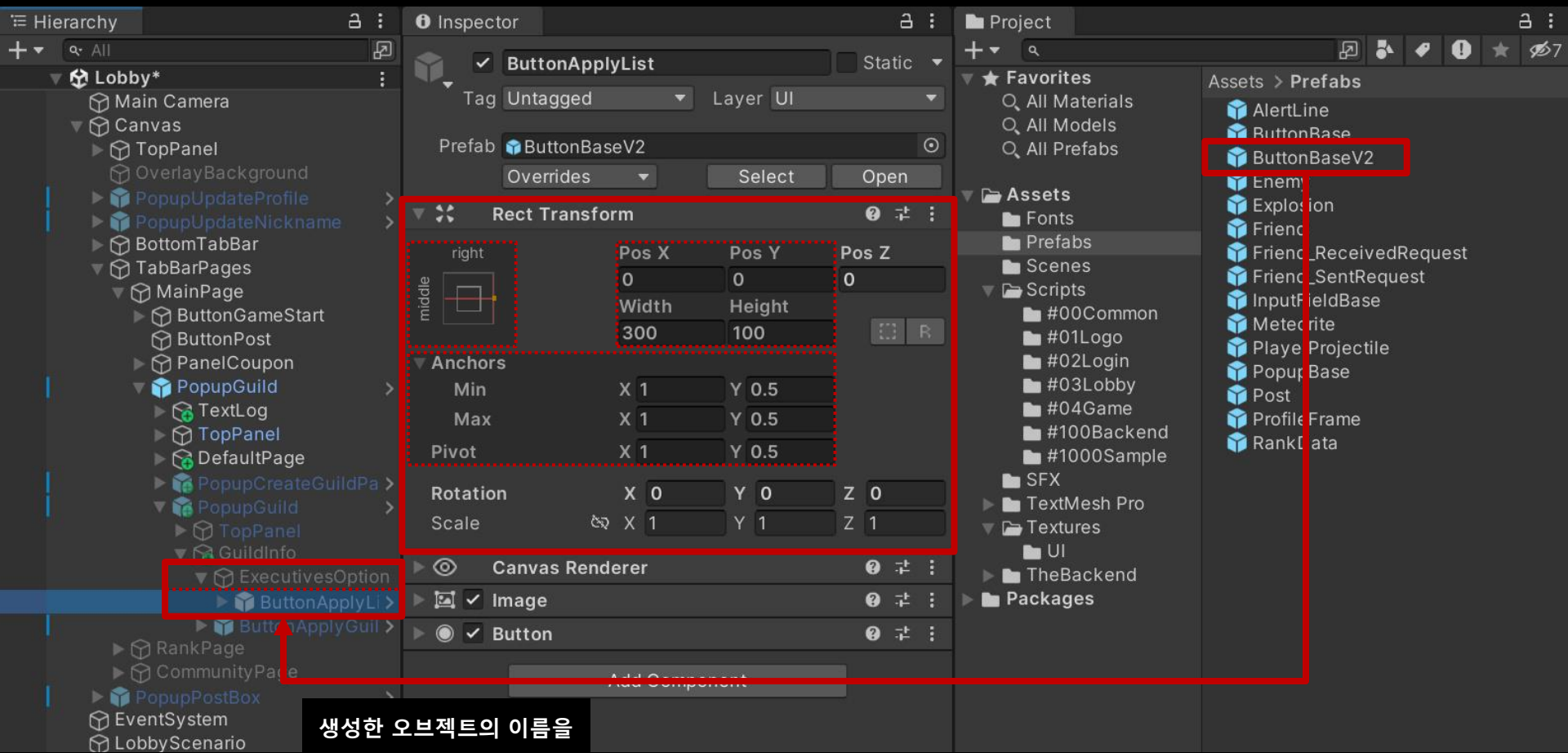
**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
      - MainPage
        - ButtonGameStart
        - ButtonPost
        - PanelCoupon
        - PopupGuild
          - TextLog
          - TopPanel
          - DefaultPage
          - PopupCreateGuildPa
          - PopupGuild
            - TopPanel
            - GuildInfo
              - ExecutivesOption
            - ButtonApplyGuil



# 길드 가입

- 길드 가입 요청 유저 목록 Button UI 생성 및 설정
  - ButtonBaseV2 프리팹을 Hierarchy View로 Drag & Drop



생성한 오브젝트의 이름을  
ButtonApplyList로 변경





# 길드 가입

- 제일 앞에 활성화되는 팝업 UI 뒤에 배치하는 배경 Panel UI 생성 및 설정
  - GameObject - UI - Panel

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree structure under 'Lobby\*'. The 'OverlayBackground' object is selected and highlighted with a red dashed box. On the right, the Inspector panel shows the 'OverlayBackground' component. The 'Rect Transform' component is highlighted with a red dashed box, showing its position and anchors. The 'Image' component is also visible, showing the 'Source Image' set to 'Background'.

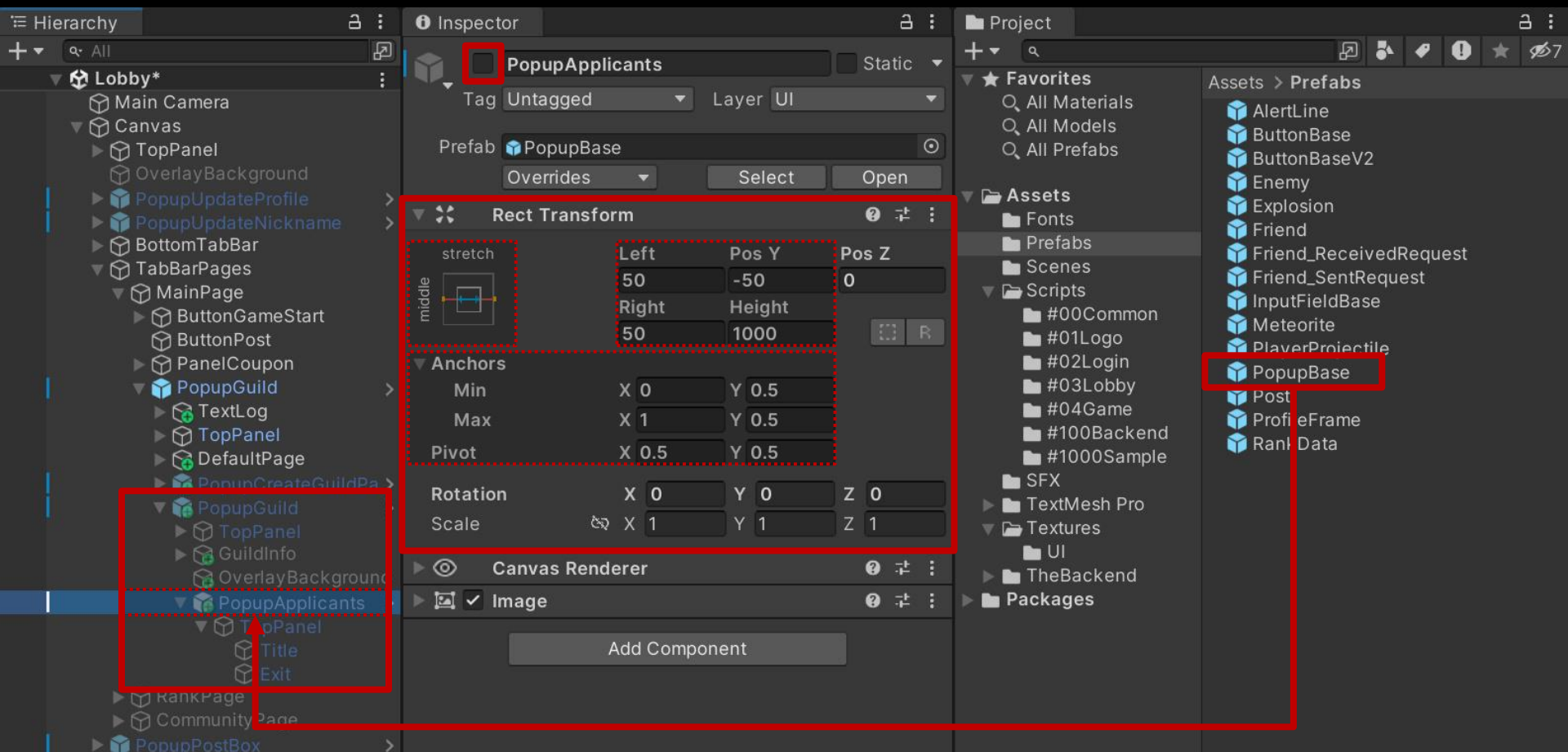
Property	Value
Tag	Untagged
Layer	UI
Stretch	stretch
Left	0
Top	100
Right	0
Bottom	0
Pos Z	0
Min X	0
Min Y	0
Max X	1
Max Y	1
Pivot X	0.5
Pivot Y	0.5
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1
Source Image	Background





# 길드 가입

- 길드 가입 요청 유저 목록 페이지 생성 및 설정
  - PopupBase 프리팹을 Hierarchy View로 Drag & Drop







# 길드 가입

- 길드 가입 요청 유저 목록 페이지 생성 및 설정 (계속)

The image shows the Unity development environment. On the left is the Hierarchy panel, and on the right is the Inspector panel.

**Hierarchy Panel:** Shows a tree structure under 'Lobby\*'. The selected object is 'Title' under 'PopupGuild'.

**Inspector Panel:** Shows the properties for the selected 'TextMeshPro - Text (UI)' component.

- Title:** Checked, Static: Unchecked
- Tag:** Untagged
- Layer:** UI
- Rect Transform:** Visible
- Canvas Renderer:** Visible
- Text Input:** Enable RTL Editor: Unchecked
- Text:** 길드 가입 요청 목록
- Text Style:** Normal
- Main Settings:**
  - Font Asset:** NotoSansKR-Bold SDF (TMP\_Fc)
  - Material Preset:** NotoSansKR-Bold SDF Material
  - Font Style:** B I U S ab AB SC
  - Font Size:** 40
  - Auto Size:** Unchecked
  - Vertex Color:** [Color Picker]
  - Color Gradient:** Unchecked



# 길드 가입

- Exit 오브젝트의 "Button" 컴포넌트 onClick 이벤트 설정

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the 'Exit' object selected under 'Lobby\*' > 'Canvas' > 'OverlayBackground'. The Inspector panel on the right shows the properties of the selected 'Exit' object, which is a Button component. The 'On Click ()' section is expanded, showing two event entries. The first entry is 'Runtime Only' with the method 'GameObject.SetActive' and the target 'OverlayBa'. The second entry is 'Runtime Only' with the method 'GameObject.SetActive' and the target 'PopupApp'. Red dashed boxes highlight the event entries and their targets. Red arrows point from the 'OverlayBa' and 'PopupApp' targets in the event list to their corresponding objects in the Hierarchy panel.



# 길드 가입

- 길드 가입 요청 유저 목록을 출력하는 Scroll View UI 생성 및 설정
  - GameObject - UI - Scroll View

The image shows the Unity development environment with the Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree view of the scene objects, with 'ApplicantsList' selected under 'Lobby\*' > 'Canvas' > 'PopupGuild' > 'ApplicantsList'. The Inspector panel on the right shows the properties for the 'ApplicantsList' object, which is a 'Static' UI element with 'Tag' set to 'Untagged' and 'Layer' set to 'UI'. The 'Rect Transform' component is highlighted with a red dashed box, showing its 'stretch' property set to a square with a crosshair, and its 'Anchors' set to 'Min X: 0, Y: 0' and 'Max X: 1, Y: 1'. The 'Canvas Renderer' component is also visible. The 'Scroll Rect' component is highlighted with a red dashed box, showing its 'Content' set to 'Content (Rect Transform)', 'Horizontal' scroll disabled, 'Vertical' scroll enabled, and 'Movement Type' set to 'Elastic'.



# 길드 가입

- Scrollbar가 보이지 않도록 Width를 0으로 설정

The image shows the Unity Inspector window with the following components and settings:

- Hierarchy:** Lobby\* > Canvas > TopPanel > Scrollbar Vertical (highlighted in red)
- Inspector:**
  - Scrollbar Vertical:** Tag: Untagged, Layer: UI
  - Rect Transform:** (highlighted in red)
    - right: Pos X: 0, Top: 0, Pos Z: 0
    - stretch: Width: 0 (highlighted with a red dashed box), Bottom: 17
    - Min: X 1, Y 0
    - Max: X 1, Y 1
    - Pivot: X 1, Y 1
    - Rotation: X 0, Y 0, Z 0
    - Scale: X 1, Y 1, Z 1
  - Canvas Renderer**
  - Image**
  - Scrollbar**

An "Add Component" button is visible at the bottom of the Inspector window.



# 길드 가입

- Content 오브젝트에 컴포넌트 추가 및 설정

The screenshot displays the Unity Inspector interface for a 'Content' object. The Hierarchy panel on the left shows a tree structure with 'Content' selected at the bottom. The Inspector panel on the right shows the 'Content' component with 'Rect Transform' and 'Vertical Layout Group' components. The 'Vertical Layout Group' component is expanded, showing 'Padding' (Left: 20, Right: 20, Top: 20, Bottom: 20, Spacing: 20, Child Alignment: Upper Center), 'Control Child Size' (Width checked, Height unchecked), 'Use Child Scale' (Width unchecked, Height unchecked), and 'Child Force Expand' (Width checked, Height checked). The 'Content Size Fitter' component is also expanded, showing 'Horizontal Fit' set to 'Unconstrained' and 'Vertical Fit' set to 'Preferred Size'. A red dashed box highlights the 'Vertical Layout Group' and 'Content Size Fitter' components and their settings.





# 길드 가입

- “[길드 가입..” 텍스트를 출력하는 “Text - TextMeshPro” UI 생성 및 설정
  - GameObject - UI - “Text - TextMeshPro”

The screenshot shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree view of the game object structure, with 'PopupGuild' and 'TextSystem' highlighted. The Inspector panel on the right shows the 'TextSystem' component with its 'Rect Transform' and 'Canvas Renderer' properties. The 'Rect Transform' component is highlighted with a red dashed box, showing its 'stretch' and 'Anchors' settings. The 'Canvas Renderer' component is also visible below it.

The screenshot shows the Unity Inspector panel for the 'TextMeshPro - Text (UI)' component. The 'Text Input' field contains the text '길드 가입 요청 유저 목록이 비어있습니다.'. The 'Main Settings' section shows the 'Font Asset' set to 'NotoSansKR-Bold SDF (TMP\_Fc)' and the 'Font Size' set to 40. The 'Alignment' section shows the text is centered. The 'Wrapping' and 'Overflow' options are also visible.





# 길드 가입

- ButtonApplyList 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The screenshot displays the Unity Inspector window for a **ButtonApplyList** component. The Hierarchy panel on the left shows the object's position within the **Lobby** scene. The Inspector panel on the right shows the **Button** component's settings, which are highlighted with a red box. The **On Click ()** section is also highlighted with a red box and contains two event entries:

- Event 1: **Runtime Only** | **GameObject.SetActive** | Target: **OverlayBa** | Checked:
- Event 2: **Runtime Only** | **GameObject.SetActive** | Target: **PopupApp** | Checked:

Red arrows point from the **OverlayBa** and **PopupApp** targets in the event list to their corresponding objects in the Hierarchy panel.



# 길드 가입

- 가입 대기 중인 유저 UI를 관리하는 오브젝트 생성 및 설정
  - Friend\_ReceivedRequest 프리팹을 Ctrl+D로 복제

The screenshot displays the Unity development environment. The Hierarchy panel on the left shows a 'Guild\_Applicant' object selected under a 'Canvas (Environment)'. The Inspector panel in the center shows the 'Rect Transform' component with its stretch and anchors settings, and the 'Friend Received Request (Script)' component with its script reference set to 'FriendReceivedRequest' and its 'Text Nickname' and 'Text Time' fields populated with 'Nickname (Text Mesh Pro UGUI)' and 'ExpirationDate (Text Mesh Pro L...' respectively. The Project panel on the right shows the 'Assets > Prefabs' folder with 'Friend\_ReceivedRequest' and 'Guild\_Applicant' prefabs highlighted. A player profile for '고박사 Lv. 1' is visible in the bottom right corner.



# 길드 가입

- 닉네임을 출력하는 Text UI 설정

The screenshot shows the Unity Inspector window for a 'Nickname' text UI component. The 'Rect Transform' component is highlighted with a red dashed box. The 'Anchors' section is also highlighted with a red dashed box. The 'TextMeshPro - Text (UI)' component is visible below the 'Rect Transform' component.

Property	Value
Pos X	180
Pos Y	-10
Pos Z	0
Width	700
Height	75
Min X	0
Min Y	1
Max X	0
Max Y	1
Pivot X	0
Pivot Y	1
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1

Canvas Renderer  
TextMeshPro - Text (UI)  
NotoSansKR-Bold SDF Material (Material)  
Shader: TextMeshPro/Distance Fi



# 길드 가입

- 레벨을 출력하는 Text UI 설정

ExpirationDate 오브젝트의 이름을 Level로 변경

Property	Value
Pos X	180
Pos Y	10
Pos Z	0
Width	700
Height	75



# 길드 가입

- 레벨을 출력하는 Text UI 설정 (계속)

The screenshot displays the Unity Inspector for a TextMeshPro component. The Hierarchy panel on the left shows the object path: **Guild\_Applicant** > **Canvas (Environment)** > **Guild\_Applicant** > **ProfileFrame** > **Level**. The Inspector panel shows the following settings for the selected **TextMeshPro - Text (UI)** component:

- Text Input:** Lv. 1
- Text Style:** Normal
- Main Settings:**
  - Font Asset: NotoSansKR-Bold SDF (TMP\_Fc)
  - Material Preset: NotoSansKR-BoId SDF Material
  - Font Style: B I U S ab AB SC
  - Font Size: 36
  - Auto Size:
  - Vertex Color:
  - Color Gradient:
  - Override Tags:
  - Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0
  - Alignment:  Left
  - Wearing: Enabled



# 길드 가입

- 길드 가입 요청 유저의 UI를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildApplicant"로 변경

```
1  using UnityEngine;
2  using TMPro;
3
4  public class GuildApplicant : MonoBehaviour
5  {
6      [SerializeField]
7      private TextMeshProUGUI textNickname;
8      [SerializeField]
9      private TextMeshProUGUI textLevel;
10
11     public void Setup(GuildMemberData memberData)
12     {
13         textNickname.text = memberData.nickname;
14         textLevel.text = $"Lv. {memberData.level}";
15     }
16 }
```





# 길드 가입

- Guild\_Applicant 프리팹에 "GuildApplicant" 컴포넌트 추가 및 설정

The screenshot displays the Unity development environment. The Hierarchy panel on the left shows a 'Guild\_Applicant' object selected under a 'Canvas (Environment)'. The Inspector panel in the center shows the 'Guild Applicant (Script)' component. The 'Script' dropdown is set to 'GuildApplicant'. Below it, the 'Text Nickname' and 'Text Level' fields are assigned to 'Nickname (Text Mesh Pro UGUI)' and 'Level (Text Mesh Pro UGUI)' respectively. The 'Assets' panel on the right shows the 'Assets > Prefabs' folder containing the 'Guild\_Applicant' prefab, which is highlighted with a red box. Red arrows point from the 'Nickname' and 'Level' fields in the Inspector to the corresponding 'Nickname' and 'Level' components in the Prefabs list.



# 길드 가입

- 길드 가입 요청 유저 목록 페이지를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "GuildApplicantsPage"로 변경

```
1 using UnityEngine;
2
3 public class GuildApplicantsPage : MonoBehaviour
4 {
5     [SerializeField]
6     private BackendGuildSystem backendGuildSystem;
7     [SerializeField]
8     private GameObject applicantPrefab;
9     [SerializeField]
10    private Transform parentContent;
11    [SerializeField]
12    private GameObject textSystem; // 해당 페이지가 비어있을 때 출력하는 Text UI
13
14    private MemoryPool memoryPool;
15
```



# 길드 가입

- 길드 가입 요청 유저 목록 페이지를 제어하는 스크립트 생성 및 작성 (계속)

```
16 private void Awake()  
17 {  
18     memoryPool = new MemoryPool(applicantPrefab, parentContent);  
19 }  
20  
21 private void OnEnable()  
22 {  
23     backendGuildSystem.GetApplicants();  
24 }  
25  
26 private void OnDisable()  
27 {  
28     DeactivateAll();  
29 }  
30
```



# 길드 가입

- 길드 가입 요청 유저 목록 페이지를 제어하는 스크립트 생성 및 작성 (계속)

```
31 public void Activate(GuildMemberData applicant)
32 {
33     if ( textSystem.activeSelf ) textSystem.SetActive(false);
34
35     GameObject item = memoryPool.ActivatePoolItem();
36     item.GetComponent<GuildApplicant>().Setup(applicant);
37 }
38
39 public void Deactivate(GameObject applicant)
40 {
41     memoryPool.DeactivatePoolItem(applicant);
42
43     if ( memoryPool.ActiveCount == 0 )
44     {
45         textSystem.SetActive(true);
46     }
47 }
48
49 public void DeactivateAll()
50 {
51     textSystem.SetActive(true);
52
53     memoryPool.DeactivateAllPoolItems();
54 }
55 }
```



# 길드 가입

- PopupApplicants 오브젝트에 "GuildApplicantsPage" 컴포넌트 추가 및 설정

The screenshot displays the Unity development environment with three main panels:

- Hierarchy Panel:** Shows the scene structure. The **PopupApplicants** object is selected and highlighted with a red box. Its children include **Content** and **TextSystem**, which are also highlighted with red boxes.
- Inspector Panel:** Shows the properties of the selected **PopupApplicants** object. The **Script** component is set to **GuildApplicantsPage**. Below it, the **Backend Guild System** is set to **BackendSystem (Backend Guild)**, the **Applicant Prefab** is set to **Guild\_Applicant**, and the **Parent Content** is set to **Content (Rect Transform)**. These three settings are enclosed in a red dashed box. The **Text System** is set to **TextSystem**. A red box highlights the **BackendSystem (Backend Guild)** component in the Inspector.
- Project Panel:** Shows the **Assets > Prefabs** folder containing the **Guild\_Applicant** prefab, which is highlighted with a red box. A red arrow points from this prefab to the **Applicant Prefab** property in the Inspector.





# 길드 가입

- Activate() 메소드를 호출해 길드 가입 요청 유저 목록 UI 출력
  - BackendGuildSystem Script 수정

```
1  +using ...
5
6  public class BackendGuildSystem : MonoBehaviour
7  {
8      [SerializeField]
9      private FadeEffect_TMP      textLog;
10     [SerializeField]
11     private GuildCreatePage      guildCreatePage;
12     [SerializeField]
13     private GuildApplicantsPage  guildApplicantsPage;
14
15     + public void CreateGuild(string guildName, int goodsCount=1)...
32
33     + public void ApplyGuild(string guildName)...
51
```





# 길드 가입

- BackendGuildSystem Script 수정 (계속)

```
52 public void GetApplicants()  
53 {  
54     Backend.Guild.GetApplicantsV3(callback =>  
55     {  
56         if ( !callback.IsSuccess() ) ...  
63  
64         // JSON 데이터 파싱 성공  
65         try  
66         {  
67             LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];  
68  
69             if ( jsonData.Count <= 0 ) ...  
74  
75             // 길드 가입 요청 목록에 있는 모든 UI 비활성화  
76             guildApplicantsPage.DeactivateAll();  
77  
78             List<TransactionValue> transactionList = new List<TransactionValue>();  
79             List<GuildMemberData> guildMemberDataList = new List<GuildMemberData>();  
80  
81             foreach ( LitJson.JsonData item in jsonData ) ...  
95
```



# 길드 가입

## BackendGuildSystem Script 수정 (계속)

```
96 Backend.GameData.TransactionReadV2(transactionList, callback =>
97 {
98     if ( !callback.IsSuccess() ) ...
103
104     LitJson.JsonData userData = callback.GetFlattenJSON()["Responses"];
105
106     if ( userData.Count <= 0 ) ...
111
112     for ( int i = 0; i < userData.Count; ++ i )
113     {
114         guildMemberDataList[i].level = userData[i]["level"].ToString();
115         guildApplicantsPage.Activate(guildMemberDataList[i]);
116         Debug.Log(guildMemberDataList[i].ToString());
117     }
118     });
119 }
120 // JSON 데이터 파싱 실패
121 catch ( Exception e )
122 {
123     // try-catch 에러 출력
124     Debug.LogError(e);
125 }
126 });
127 }
```



# 길드 가입

- BackendSystem 오브젝트의 "BackendGuildSystem" 컴포넌트 변수 설정

The screenshot displays the Unity Inspector for the **BackendSystem** component. The **Backend Guild System (Script)** component is selected, and its variables are configured as follows:

- Script: BackendGuildSystem
- Text Log: TextLog (Fade Effect\_TMP)
- Guild Create Page: PopupCreateGuildPage (Guild C)
- Guild Applicants Page: PopupApplicants (Guild Applicar)

The **BackendSystem** object is highlighted in the Hierarchy panel, and the **PopupApplicants** object is also highlighted. A red arrow points from the **PopupApplicants** variable in the Inspector to the **PopupApplicants** object in the Hierarchy panel.



# 길드 가입

## ■ 결과 화면

# Scene Game Animator Animation

Game Display 1 1440x2960 Scale 0.39x Play Focused Stats Gizmos

닉네임 2 30 / 30 22:22 2734 6218

Enter text... 쿠폰등록

**길드 (Guild)**

검색할 길드명을 입력하세요. 길드 검색

길드 생성

**게임시작**

Console

Clear Collapse Error Pause Editor

UnityEngine.Debug:Log (object)

[22:36:24] Level : 9, Max Exp : 25600, Reward Gold : 9999  
UnityEngine.Debug:Log (object)

[22:36:24] Level : 10, Max Exp : 51200, Reward Gold : 100000  
UnityEngine.Debug:Log (object)

[22:36:24] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200  
message : Success

[22:36:29] nickname : user03  
inDate : 2023-06-26T06:43:54.899Z

[22:36:29] nickname : 고박사  
inDate : 2023-06-11T13:12:57.374Z



# 길드 가입

## ■ 길드 가입 요청 승인

- ApproveApplicant() 메소드를 호출해 길드 가입 요청 승인
  - BackendGuildSystem Script 수정

```
52      + public void GetApplicants()...
128
129      - public void ApproveApplicant(string gamerInDate)
130      {
131          Backend.Guild.ApproveApplicantV3(gamerInDate, callback =>
132          {
133              if ( !callback.IsSuccess() )
134              {
135                  ErrorLogApproveApplicant(callback);
136
137                  return;
138              }
139
140              Debug.Log($"길드 가입 요청 수락에 성공했습니다. : {callback}");
141          });
142      }
143
144      + public string GetGuildInfoBy(string guildName)...
169
```





# 길드 가입

- BackendGuildSystem Script 수정 (계속)

```
170 private void ErrorLogCreateGuild(BackendReturnObject callback)...
189
190 private void ErrorLogApplyGuild(BackendReturnObject callback)...
212
213 private void ErrorLogApproveApplicant(BackendReturnObject callback)
214 {
215     string message = string.Empty;
216
217     switch ( int.Parse(callback.GetStatusCode()) )
218     {
219         case 412:
220             message = "길드 가입 요청을 수락하려는 유저가 이미 다른 길드 소속입니다.";
221             break;
222         case 429:
223             message = "길드에 더 이상 자리가 없습니다.";
224             break;
225     }
226
227     ErrorLog(message, "Guild_Failed_Log", "ApproveApplicant");
228 }
229
230 private void ErrorLog(string message, string behaviorType="", string paramKey="")...
243 }
```





# 길드 가입

- 길드 가입 요청 승인 버튼을 눌렀을 때 처리
  - GuildApplicant Script 수정

```
1  using UnityEngine;
2  using TMPro;
3
4  public class GuildApplicant : MonoBehaviour
5  {
6      [SerializeField]
7      private TextMeshProUGUI    textNickname;
8      [SerializeField]
9      private TextMeshProUGUI    textLevel;
10
11     private BackendGuildSystem  backendGuildSystem;
12     private GuildApplicantsPage guildApplicantsPage;
13     private GuildMemberData    guildMemberData;
14
```



# 길드 가입

## □ GuildApplicant Script 수정 (계속)

```
15 public void Setup(BackendGuildSystem guildSystem, GuildApplicantsPage applicantsPage, GuildMemberData memberData)
16 {
17     textNickname.text = memberData.nickname;
18     textLevel.text = $"Lv. {memberData.level}";
19
20     backendGuildSystem = guildSystem;
21     guildApplicantsPage = applicantsPage;
22     guildMemberData = memberData;
23 }
24
25 public void OnClickApproveApplicant()
26 {
27     // 해당 유저의 UI 오브젝트 삭제
28     guildApplicantsPage.Deactivate(gameObject);
29     // 해당 유저 길드 가입 요청 수락
30     backendGuildSystem.ApproveApplicant(guildMemberData.inDate);
31 }
32 }
```



# 길드 가입

- 길드 가입 요청 유저 UI를 생성할 때 GuildApplicant.Setup() 메소드 호출
  - GuildApplicantsPage Script 수정

```
1  using UnityEngine;
2
3  public class GuildApplicantsPage : MonoBehaviour
4  {
5      [SerializeField]
6      private BackendGuildSystem backendGuildSystem;
7      [SerializeField]
8      private GameObject applicantPrefab;
9      [SerializeField]
10     private Transform parentContent;
11     [SerializeField]
12     private GameObject textSystem; // 해당 페이지가 비어있을 때 출력하는 Text UI
13
14     private MemoryPool memoryPool;
15
16     private void Awake()...
17
18
19
20
21     private void OnEnable()...
22
23
24
25
26     private void OnDisable()...
27
28
29
30
31     public void Activate(GuildMemberData applicant)
32     {
33         if ( textSystem.activeSelf ) textSystem.SetActive(false);
34
35         GameObject item = memoryPool.ActivatePoolItem():
36         item.GetComponent<GuildApplicant>().Setup(backendGuildSystem, this, applicant);
37     }
38
39     public void Deactivate(GameObject applicant)...
40
41
42
43
44
45
46
47
48
49     public void DeactivateAll()...
50
51
52
53
54
55 }
```



# 길드 가입

- ButtonAccept 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The screenshot shows the Unity Inspector for a **ButtonAccept** component. The **Button** sub-component is selected, and its **On Click ()** event is configured. The event is set to **Runtime Only** and the method **GuildApplicant.OnClickApproveAp** is selected. The **Inspector** panel shows the following properties for the **Button** component:

- Interactable:
- Transition: Color Tint
- Target Graphic: ButtonAccept (Image)
- Normal Color: [Color Picker]
- Highlighted Color: [Color Picker]
- Pressed Color: [Color Picker]
- Selected Color: [Color Picker]
- Disabled Color: [Color Picker]
- Color Multiplier: 1
- Fade Duration: 0.1
- Navigation: Automatic
- Visualize: [Button]
- On Click ():
  - Runtime Only
  - GuildApplicant.OnClickApproveAp
  - Guild\_Appl



# 길드 가입

## ■ 결과 화면

Unity Game View showing the Guild Join screen. The UI includes a top status bar with player name, level (30/30), and resources. Below is an input field for text and a ' Guild Search' button. A ' Guild Search' button is also present. At the bottom, there is a ' Guild Create' button and a large yellow ' Start Game ' button. The console on the right shows the following logs:

```

[22:55:15] Level : 10, Max Exp : 51200,Reward Gold : 100000
UnityEngine.Debug.Log (object)
[22:55:15] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200
message : Success
[22:55:21] nickname : user03
inDate : 2023-06-26T06:43:54.899Z
[22:55:21] nickname : 고박사
inDate : 2023-06-11T13:12:57.374Z
[22:55:23] 길드 가입 요청 수락에 성공했습니다. : statusCode : 204
message : Success
길드 가입 요청 수락에 성공했습니다. : statusCode : 204
message : Success
UnityEngine.Debug.Log (object)
BackendGuildSystem:<ApproveApplicant>b__6_0 (BackEnd.BackendReturnO
#W.#V:#U ()
  
```



# 길드 가입

## ■ 결과 화면 (Backend Console)

Backend Console ProjectA

길드 관리 삭제

전체 길드명으로 검색

SDK 문서 콘솔 가이드

길드 설정

번호 길드명 inDate 생성일 국가 길드마스터 닉네임 인원 굿즈 메타

**유니티노트 길드원 목록**

삭제

번호	유저 번호	닉네임	가입일	최종 접속일	등급
1	aedaa3e0-0859-11ee-8b11-73576fc879f3	교박사	2023.07.12 22:55	2023.07.12 16:13	길드원
2	b9981f10-0859-11ee-9227-0f2b793d8148	닉네임	2023.07.11 16:15	2023.07.12 22:55	길드마스터

< 1 >

10개씩 보기

확인

회사소개 이용약관 서비스수준협약 개인정보처리방침 © AFI, Inc. All rights reserved.





# 길드 가입

## ■ 길드 가입 요청 거절

- RejectApplicant() 메소드를 호출해 길드 가입 요청 거절

- BackendGuildSystem Script 수정

```
52      public void GetApplicants()...
128
129      public void ApproveApplicant(string gamerInDate)...
143
144      public void RejectApplicant(string gamerInDate)
145      {
146          Backend.Guild.RejectApplicantV3(gamerInDate, callback =>
147          {
148              if ( !callback.IsSuccess() )
149              {
150                  // 운영진에게만 해당 버튼이 보일 예정으로
151                  // 실패 사유가 404 하나 밖에 없기 때문에 별도로 메소드 제작 x
152                  ErrorLog(callback.GetMessage(), "Guild_Failed_Log", "RejectApplicant");
153
154                  return;
155              }
156
157              Debug.Log($"길드 가입 요청 거절에 성공했습니다. : {callback}");
158          });
159      }
160
161      public string GetGuildInfoBy(string guildName)...
```



# 길드 가입

- 길드 가입 요청 거절 버튼을 눌렀을 때 처리
  - GuildApplicant Script 수정

```
1  using UnityEngine;
2  using TMPro;
3
4  public class GuildApplicant : MonoBehaviour
5  {
6      [SerializeField]
7      private TextMeshProUGUI textNickname;
8      [SerializeField]
9      private TextMeshProUGUI textLevel;
10
11     private BackendGuildSystem backendGuildSystem;
12     private GuildApplicantsPage guildApplicantsPage;
13     private GuildMemberData guildMemberData;
14
15     public void Setup(BackendGuildSystem guildSystem, GuildApplicantsPage applicantsPage, GuildMemberData memberData) ...
24
25     public void OnClickApproveApplicant() ...
32
33     public void OnClickRejectApplicant()
34     {
35         // 해당 유저의 UI 오브젝트 삭제
36         guildApplicantsPage.Deactivate(gameObject);
37         // 해당 유저 길드 가입 요청 거절
38         backendGuildSystem.RejectApplicant(guildMemberData.inDate);
39     }
40 }
```



# 길드 가입

- ButtonReject 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The screenshot displays the Unity Inspector for a **ButtonReject** component. The Hierarchy panel on the left shows the object path: **Guild\_Applicant** > **ButtonReject**. The Inspector panel shows the following settings for the **Button** component:

- Interactable:**
- Transition:** Color Tint
- Target Graphic:** ButtonReject (Image)
- Normal Color:** [Color field]
- Highlighted Color:** [Color field]
- Pressed Color:** [Color field]
- Selected Color:** [Color field]
- Disabled Color:** [Color field]
- Color Multiplier:** 1
- Fade Duration:** 0.1
- Navigation:** Automatic

The **On Click ()** event is configured with the following settings:

- Runtime Only:**
- Event Name:** GuildApplicant.OnClickRejectApp
- Target:** Guild\_Appl



# 길드 가입

## ■ 결과 화면

Unity Editor interface showing a game scene with a guild search UI and a console log.

**Game UI Elements:**

- Top bar: # Scene, Game, Animator, Animation
- Game View: Display 1, 1440x2960, Scale 0.39x, Play Focused
- Player Info: 닉네임 2, HP 30/30, EXP 2734, Gold 6218
- Input Field: Enter text... (with envelope icon)
- Buttons: 쿨온등록, 길드 검색, 길드 생성
- Bottom Bar: 게임시작 (large yellow button)
- Bottom Icons: Shop, Backpack, Swords, Leaderboard, Profile

**Console Log (Right Panel):**

```

[23:02:56] Level : 9, Max Exp : 25600,Reward Gold : 9999
UnityEngine.Debug:Log (object)

[23:02:56] Level : 10, Max Exp : 51200,Reward Gold : 100000
UnityEngine.Debug:Log (object)

[23:02:56] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200
message : Success

[23:03:03] nickname : user04
inDate : 2023-06-26T06:45:15.098Z

[23:03:05] 길드 가입 요청 거절에 성공했습니다. : statusCode : 204
message : Success
  
```