

Unity Backend

“친구” 기능을 이용한 친구 수락 대기 목록, 친구 목록

Created in 2023-07-05
Last Updated 2023-07-17
Unity Version 2022.2.2f1

Index

- ◆ 친구 수락 대기 목록
- ◆ 친구 목록

친구 수락 대기 목록

- 친구 수락 대기 목록 조회
- 친구 수락 대기 목록 UI 제작
- 친구 수락 대기 목록 UI 출력
- 친구 수락
- 친구 거절



친구 수락 대기 목록

■ 친구 수락 대기 목록 조회

■ 친구 수락 대기 목록 조회 메소드

```
GetReceivedRequestList(int limit=100, int offset=0);
```

친구 요청을 받은 목록 조회

limit : 불러올 친구 수락 대기 목록의 수 (default : 100)

offset : 불러올 친구 수락 대기 기준 숫자

ex) GetReceivedRequestList(5); -> 친구 요청을 받은 5명 목록 조회 (1-5)

ex) GetReceivedRequestList(5, 5); -> 친구 요청을 받은 처음 5명 이후의 5명 조회 (6-10)



친구 수락 대기 목록

■ 친구 수락 대기 목록 JsonData

```
{
  "rows":
  [
    // 닉네임이 있는 유저
    {
      "nickname": "고박사" // 유저의 닉네임
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "createdAt": "2023-06-26T02:22:22.022Z" // 친구 요청 받은 시간
    },
    // 닉네임이 없는 유저
    {
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "createdAt": "2023-06-26T02:22:22.022Z" // 친구 요청 받은 시간
    },
  ]
}
```



친구 수락 대기 목록

- GetReceivedRequestList() 메소드를 호출해 친구 [수락대기] 목록 조회
 - BackendFriendSystem Script 수정

```
1  +using ...
4
5  public class BackendFriendSystem : MonoBehaviour
6  {
7      [SerializeField]
8      private FriendSentRequestPage    sentRequestPage;
9
10     +private string GetUserInfoBy(string nickname)...
47
48     +public void SendRequestFriend(string nickname)...
68
69     +public void GetSentRequestList()...
122
123    +public void RevokeSentRequest(string inDate)...
136
137    public void GetReceivedRequestList()
138    {
139        Backend.Friend.GetReceivedRequestList(callback =>
140        {
141            if ( !callback.IsSuccess() )
142            {
143                Debug.LogError($"친구 수락 대기 목록 조회 도중 에러가 발생했습니다. : {callback}");
144                return;
145            }
146        }
```



친구 수락 대기 목록

BackendFriendSystem Script 수정 (계속)

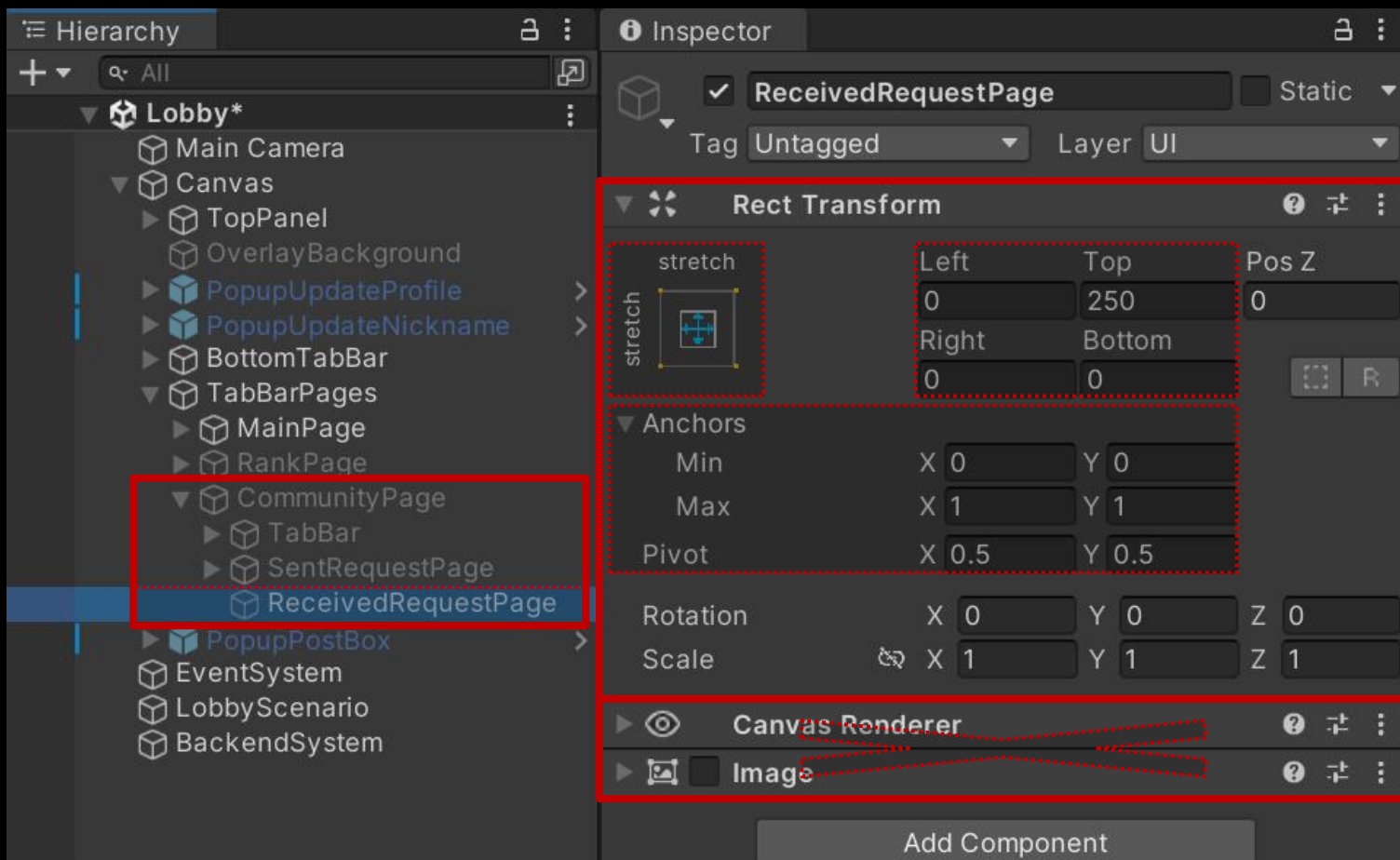
```
147 // JSON 데이터 파싱 성공
148 try
149 {
150     LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];
151
152     // 받은 데이터의 개수가 0이면 데이터가 없는 것
153     if ( jsonData.Count <= 0 )
154     {
155         Debug.LogWarning("친구 수락 대기 목록 데이터가 없습니다.");
156         return;
157     }
158
159     foreach ( LitJson.JsonData item in jsonData )
160     {
161         FriendData friendData = new FriendData();
162
163         friendData.nickname = item["nickname"].ToString().Equals("True") ? "NONAME" : item["nickname"].ToString();
164         friendData.inDate = item["inDate"].ToString();
165         friendData.createdAt = item["createdAt"].ToString();
166     }
167
168     // JSON 데이터 파싱 실패
169     catch ( Exception e )
170     {
171         // try-catch 에러 출력
172         Debug.LogError(e);
173     }
174 });
175 }
176
177 private bool IsExpirationDate(string createdAt){...
212 }
```



친구 수락 대기 목록

친구 수락 대기 목록 UI 제작

- 친구 수락 페이지를 관리하는 Panel UI 생성 및 설정
 - GameObject - UI - Panel





친구 수락 대기 목록

- 친구 수락 대기 목록을 출력하는 Scroll View UI 생성 및 설정
 - GameObject - UI - Scroll View

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby*' with 'ReceivedRequestList' selected. The Inspector panel on the right shows the properties for 'ReceivedRequestList', which is a 'Rect Transform' component. The 'Rect Transform' component is highlighted with a red dashed box and contains the following settings:

Property	Value
Left	0
Top	0
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

Below the 'Rect Transform' component, the 'Canvas Renderer' and 'Image' components are visible. The 'Image' component is also highlighted with a red dashed box. The 'Scroll Rect' component is visible at the bottom of the Inspector panel, with the 'Content' property set to 'Content (Rect Transform)'. The 'Horizontal' checkbox is checked, and the 'Vertical' checkbox is unchecked.



친구 수락 대기 목록

- Scrollbar가 보이지 않도록 Width를 0으로 설정

The image shows the Unity Inspector window for a 'Scrollbar Vertical' component. The 'Rect Transform' section is highlighted with a red box, and the 'Width' field is also highlighted with a red dashed box. The 'Width' is set to 0, and the 'Bottom' is set to 17. The 'Anchors' section shows 'Min' at (X: 1, Y: 0) and 'Max' at (X: 1, Y: 1). The 'Rotation' and 'Scale' sections are also visible.

Property	X	Y	Z
Pos X	0		0
Top		0	
Pos Z			0
Width	0		
Bottom		17	
Min X	1		
Min Y		0	
Max X	1		
Max Y		1	
Pivot X	1		
Pivot Y		1	
Rotation X	0		0
Rotation Y		0	
Rotation Z			0
Scale X	1		
Scale Y		1	
Scale Z			1



친구 수락 대기 목록

- Content 오브젝트에 컴포넌트 추가 및 설정

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby*'. The 'Content' object is selected and highlighted with a red box. The Inspector panel on the right shows the 'Content' component with the following settings:

- Content** (checked)
- Tag: Untagged
- Layer: UI
- Rect Transform** (checked)
- Vertical Layout Group** (checked)
 - Padding: Left (20), Right (20), Top (20), Bottom (20)
 - Spacing: 20
 - Child Alignment: Upper Center
 - Reverse Arrangement:
 - Control Child Size: Width, Height
 - Use Child Scale: Width, Height
 - Child Force Expand: Width, Height
- Content Size Fitter** (checked)
 - Horizontal Fit: Unconstrained
 - Vertical Fit: Preferred Size

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



친구 수락 대기 목록

- “[친구 수락..” 텍스트를 출력하는 “Text - TextMeshPro” UI 생성 및 설정
 - GameObject - UI - “Text - TextMeshPro”

Hierarchy: Lobby* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > MainPage > RankPage > CommunityPage > TabBar > SentRequestPage > ReceivedRequestPage > ReceivedRequestList > TextSystem

Inspector: TextSystem (Tag: Untagged, Layer: UI)

Rect Transform: stretch, middle, Left: 0, Pos Y: 0, Pos Z: 0, Right: 0, Height: 50

Anchors: Min (X: 0, Y: 0.5), Max (X: 1, Y: 0.5), Pivot (X: 0.5, Y: 0.5)

Rotation: X: 0, Y: 0, Z: 0

Scale: X: 1, Y: 1, Z: 1

Canvas Renderer

TextMeshPro - Text (UI)

Text Input: [친구 수락 대기] 목록이 비어있습니다.

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: 60

Auto Size:

Vertex Color:

Color Gradient:

Override Tags:

Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0

Alignment: Center

Wrapping: Enabled

Overflow: Overflow

Horizontal Mapping: Character

Vertical Mapping: Character



친구 수락 대기 목록

- ReceivedRequestList 오브젝트의 "Toggle" 컴포넌트 OnValueChanged() 이벤트 등록

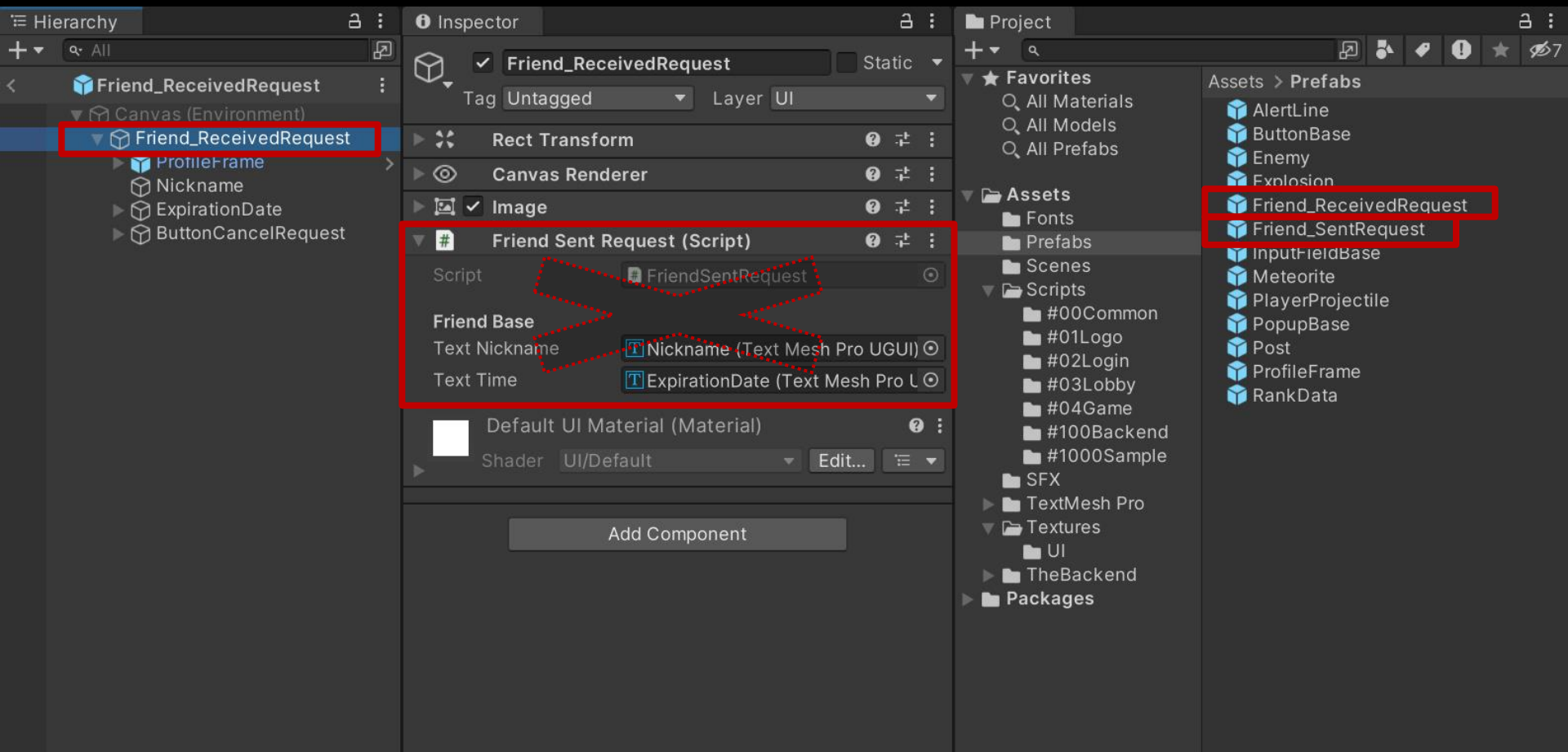
The screenshot shows the Unity Inspector window with the following configuration for the **Toggle** component:

- Interactable:**
- Transition:** Color Tint
 - Target Graphic: Background (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
- Is On:**
- Toggle Transition:** Fade
 - Graphic: Checkmark (Image)
 - Group: TabBar (Toggle Group)
- On Value Changed (Boolean):**
 - Event Type: Runtime Only
 - Method: GameObject.SetActive
 - Target: ReceivedRequestList



친구 수락 대기 목록

- [수락 대기] 중인 친구 UI를 관리하는 오브젝트 생성 및 설정
 - Friend_SentRequest 프리팹을 Ctrl+D로 복제





친구 수락 대기 목록

- 수락 대기 만료까지 남은 시간을 출력하는 Text UI 설정

The screenshot shows the Unity Inspector window for the 'ExpirationDate' component. The 'Rect Transform' component is highlighted with a red dashed box. The 'stretch' property is set to 'middle'. The 'Left' and 'Right' values are 600 and 340, respectively. The 'Pos Y' and 'Height' values are 0 and 75. The 'Anchors' section shows 'Min' at X 0, Y 0.5 and 'Max' at X 1, Y 0.5. The 'Pivot' is set to X 0.5, Y 0.5. The 'Rotation' and 'Scale' properties are also visible.

Property	Value
Left	600
Right	340
Pos Y	0
Height	75
Min X	0
Min Y	0.5
Max X	1
Max Y	0.5
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



친구 수락 대기 목록

□ "거절" 버튼 설정

The screenshot shows the Unity Inspector window for a **ButtonReject** component. The **Rect Transform** component is highlighted with a red dashed box. The **Position** (Pos X, Pos Y, Pos Z) and **Size** (Width, Height) values are also highlighted with a red dashed box. The **Anchor** settings (Min, Max, Pivot) are also visible.

Property	Value
Pos X	-10
Pos Y	0
Pos Z	0
Width	150
Height	150
Min X	1
Min Y	0.5
Max X	1
Max Y	0.5
Pivot X	1
Pivot Y	0.5
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1

ButtonCancelRequest 오브젝트의 이름을 **ButtonReject**로 변경



친구 수락 대기 목록

- "거절" 버튼 설정 (계속)

The screenshot displays the Unity Inspector window for a **Text (TMP)** component. The **TextMeshPro - Text (UI)** sub-component is selected and highlighted with a red box. The text content is "거절". The **Main Settings** section is visible, showing the following configuration:

- 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:
- Override Tags:

The Hierarchy panel on the left shows the object structure, with **Text (TMP)** selected under **ButtonReject**. The Project panel on the right shows the **Assets** folder structure, including **TextMesh Pro** and **UI** sub-folders.



친구 수락 대기 목록

□ "수락" 버튼 설정

The screenshot shows the Unity Inspector for a **ButtonAccept** component. The **Rect Transform** component is highlighted with a red dashed box. The properties are as follows:

Property	Value
Pos X	-170
Pos Y	0
Pos Z	0
Width	150
Height	150
Min X	1
Min Y	0.5
Max X	1
Max Y	0.5
Pivot X	1
Pivot Y	0.5
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1

The Hierarchy panel on the left shows the following structure:

- Friend_ReceivedRequest
 - Canvas (Environment)
 - Friend_ReceivedRequest
 - ProfileFrame
 - Nickname
 - ExpirationDate
 - ButtonReject
 - ButtonAccept** (highlighted)
 - Text (TMP)

The Project panel on the right shows the following structure:

- Assets
 - Fonts
 - Prefabs
 - Scenes
 - Scripts
 - #00Common
 - #01Logo
 - #02Login
 - #03Lobby
 - #04Game
 - #100Backend
 - #1000Sample
 - SFX
 - TextMesh Pro
 - Textures
 - UI
 - TheBackend
- Packages

The Inspector panel shows the following components for the **ButtonAccept** object:

- Image
- Button
- Default UI Material (Material)
- Shader: UI/Default

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

ButtonReject 오브젝트를 Ctrl+D로 복제한 후 이름을 ButtonAccept로 변경



친구 수락 대기 목록

- "수락" 버튼 설정 (계속)

The screenshot displays the Unity Inspector window for a TextMeshPro component. The Hierarchy panel on the left shows the object path: Friend_ReceivedRequest > Canvas (Environment) > Friend_ReceivedRequest > ProfileFrame > Text (TMP). The Inspector panel shows the following settings:

- Text (TMP)** (Static)
- Tag: Untagged, Layer: UI
- Rect Transform
- Canvas Renderer
- TextMeshPro - Text (UI)**
 - Text Input: 수락 (highlighted with a red dashed box)
 - 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:
 - Vertex Color:
 - Color Gradient:
 - Override Tags:

The Project panel on the right shows the Assets folder structure, including Fonts, Prefabs, Scenes, Scripts, and Textures.



친구 수락 대기 목록

■ 친구 수락 대기 목록 UI 출력

- 친구 수락 페이지를 제어하는 스크립트 생성 및 작성
 - C# Script 생성 후 스크립트의 이름을 "FriendReceivedRequestPage"로 변경

```
1 public class FriendReceivedRequestPage : FriendPageBase
2 {
3     private void OnEnable()
4     {
5         // [친구 수락 대기] 목록 불러오기
6         backendFriendSystem.GetReceivedRequestList();
7     }
8
9     private void OnDisable()
10    {
11        DeactivateAll();
12    }
13 }
```



친구 수락 대기 목록

- ReceivedRequestPage 오브젝트의 "FriendReceivedRequestPage" 컴포넌트 변수 설정

The screenshot displays the Unity Inspector window for the `ReceivedRequestPage` object. The `Friend Received Request Page (Script)` component is selected, and its variables are being configured:

- `Backend Friend System` is set to `BackendSystem (Backend Friend System)`.
- `Friend Page Base` is set to `Friend_ReceivedRequest`.
- `Parent Content` is set to `Content (Rect Transform)`.
- `Text System` is set to `TextSystem`.

Red boxes and arrows highlight these configurations. In the Hierarchy panel, the `ReceivedRequestPage` object and its children (`Content`, `TextSystem`) are also highlighted. In the Project panel, the `Assets > Prefabs > Friend_ReceivedRequest` prefab is highlighted.



친구 수락 대기 목록

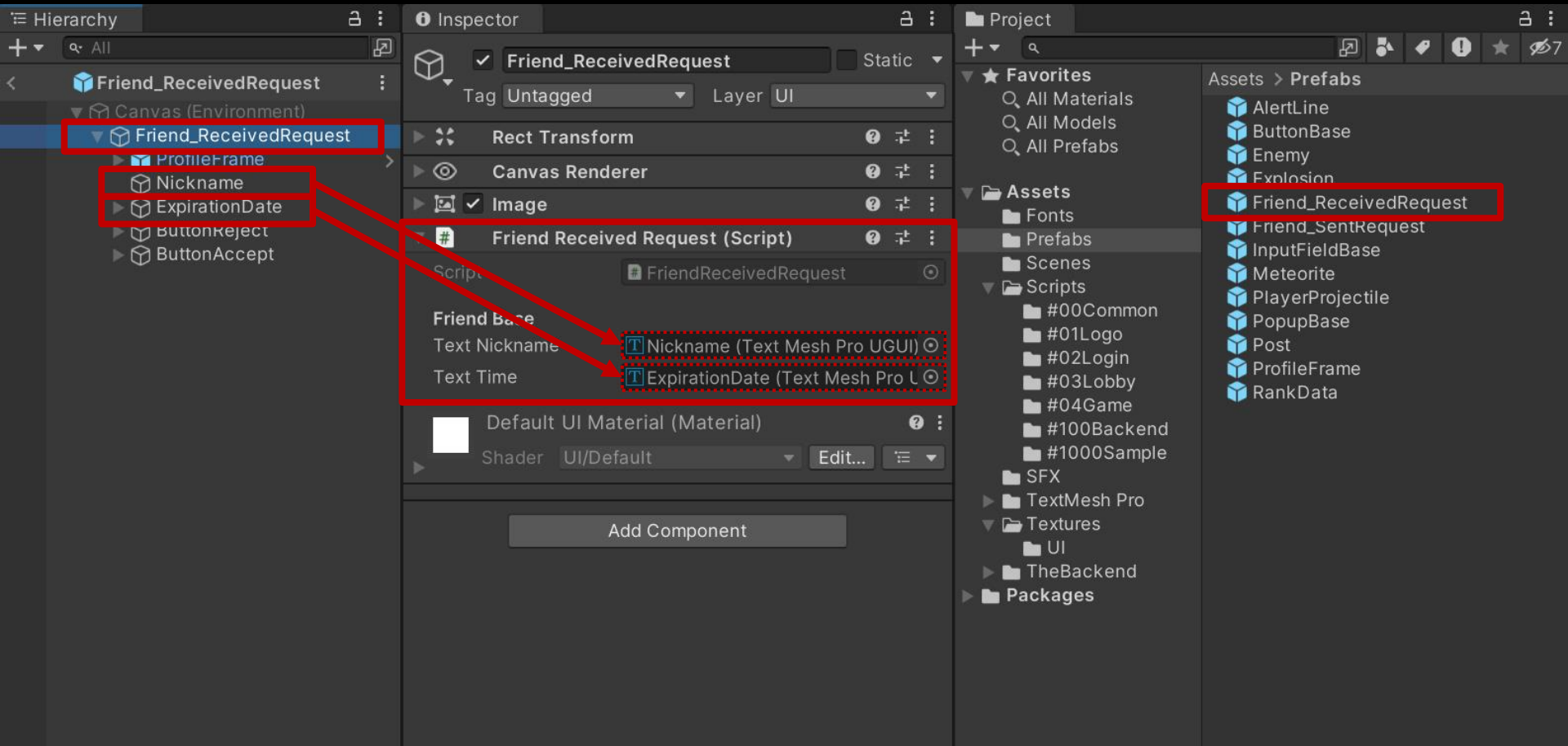
- 수락 대기 중인 친구 UI를 제어하는 스크립트 생성 및 작성
 - C# Script 생성 후 스크립트의 이름을 "FriendReceivedRequest"로 변경

```
1 public class FriendReceivedRequest : FriendBase
2 {
3     public override void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData)
4     {
5         base.Setup(friendSystem, friendPage, friendData);
6         base.SetExpirationDate();
7     }
8 }
```




친구 수락 대기 목록

- Friend_ReceivedRequest 프리팹에 "FriendReceivedRequest" 컴포넌트 추가 및 설정





친구 수락 대기 목록

- receivedRequestPage.Activate() 메소드를 호출해 친구 수락 대기 목록 UI 출력
 - BackendFriendSystem Script 수정

```
1  + using ...
4
5  - public class BackendFriendSystem : MonoBehaviour
6  {
7      [SerializeField]
8      private FriendSentRequestPage    sentRequestPage;
9      [SerializeField]
10     private FriendReceivedRequestPage receivedRequestPage;
11
12     + private string GetUserInfoBy(string nickname)...
49
50     + public void SendRequestFriend(string nickname)...
70
71     + public void GetSentRequestList()...
124
125    + public void RevokeSentRequest(string inDate)...
138
```




친구 수락 대기 목록

BackendFriendSystem Script 수정 (계속)

```
139 public void GetReceivedRequestList()
140 {
141     Backend.Friend.GetReceivedRequestList(callback =>
142     {
143         if ( !callback.IsSuccess() )...
144
145         // JSON 데이터 파싱 성공
146         try
147         {
148             LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];
149
150             // 받은 데이터의 개수가 0이면 데이터가 없는 것
151             if ( jsonData.Count <= 0 )...
152
153             // 친구 수락 대기 목록에 있는 모든 UI 비활성화
154             receivedRequestPage.DeactivateAll();
155
156             foreach ( LitJson.JsonData item in jsonData )
157             {
158                 FriendData friendData = new FriendData();
159
160                 friendData.nickname = item["nickname"].ToString().Equals("True") ? "NONAME" : item["nickname"].ToString();
161                 friendData.inDate = item["inDate"].ToString();
162                 friendData.createdAt = item["createdAt"].ToString();
163
164                 // 현재 friendData 정보를 바탕으로 친구 수락 대기 UI 활성화
165                 receivedRequestPage.Activate(friendData);
166             }
167         }
168         // JSON 데이터 파싱 실패
169         catch ( Exception e )
170         {
171             // try-catch 에러 출력
172             Debug.LogError(e);
173         }
174     });
175 }
176
177
178
179
180
181
182
183
```



친구 수락 대기 목록

- BackendSystem 오브젝트의 "BackendFriendSystem" 컴포넌트 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **BackendSystem** object is selected at the bottom. A red box highlights the **ReceivedRequestPage** component under the **CommunityPage** sub-object. In the Inspector panel, the **BackendSystem** component is selected, and its **Backend Friend System (Script)** sub-component is expanded. A red box highlights the **Received Request Page** property, which is set to **ReceivedRequestPage (Friend R...)**. A red arrow points from the **ReceivedRequestPage** component in the Hierarchy panel to the **Received Request Page** property in the Inspector panel.



친구 수락 대기 목록

■ 결과 화면

The screenshot displays a Unity game interface. At the top, the scene is set to 'Game' with a resolution of 1440x2960 and a scale of 0.39x. The game view shows a dark screen with a top status bar containing a character icon, a heart icon with '35 / 30', a diamond icon with '6936', and a stack of coins icon with '9610'. Below the status bar is an input field with the placeholder text 'Enter text...' and a yellow button labeled '쿠폰등록'. A large yellow button labeled '게임시작' is centered on the screen. At the bottom, there is a navigation bar with five icons: a shop, a backpack, crossed swords, a leaderboard, and a profile.

The console on the right shows the following logs:

- UnityEngine.Debug:LogWarning (object)
- [15:22:29] 닉네임 : user04
inDate : 2023-06-26T06:45:15.098Z
- [15:22:29] 닉네임 : 닉네임
inDate : 2023-06-11T13:13:15.393Z
- [15:22:31] 친구 요청 대기 목록 데이터가 없습니다.
UnityEngine.Debug:LogWarning (object)
- [15:22:31] 닉네임 : user04
inDate : 2023-06-26T06:45:15.098Z
- [15:22:31] 닉네임 : 닉네임
inDate : 2023-06-11T13:13:15.393Z



친구 수락 대기 목록

■ 친구 수락

- AcceptFriend() 메소드를 호출해 친구 수락
 - BackendFriendSystem Script 수정

```
125 public void RevokeSentRequest(string inDate) ...
138
139 public void GetReceivedRequestList() ...
184
185 public void AcceptFriend(FriendData friendData)
186 {
187     Backend.Friend.AcceptFriend(friendData.inDate, callback =>
188     {
189         if ( !callback.IsSuccess() )
190         {
191             Debug.LogError($"친구 수락 중 에러가 발생했습니다. : {callback}");
192             return;
193         }
194
195         Debug.Log($"{friendData.nickname}이(가) 친구가 되었습니다. : {callback}");
196     });
197 }
198
199 private bool IsExpirationDate(string createdAt) ...
234 }
```



친구 수락 대기 목록

- “친구수락” 버튼을 눌렀을 때 호출할 `OnClickAcceptRequest()` 메소드 정의
 - `FriendReceivedRequest` Script 수정

```
1 public class FriendReceivedRequest : FriendBase
2 {
3     public override void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData) ...
8
9     public void OnClickAcceptRequest()
10 {
11     // 친구 UI 오브젝트 삭제
12     friendPage.Deactivate(gameObject);
13     // 친구 요청 수락 (Backend Console)
14     backendFriendSystem.AcceptFriend(friendData);
15 }
16 }
```



친구 수락 대기 목록

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

The screenshot displays the Unity Inspector window for a Button component attached to a ButtonAccept object. The Hierarchy panel on the left shows the object structure, with 'Friend_ReceivedRequest' and 'ButtonAccept' highlighted. The Inspector panel shows the following configuration for the Button component:

- Interactable:
- Transition: Color Tint
- Target Graphic: ButtonAccept (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 'OnClick ()' event list is visible at the bottom, showing a single event:

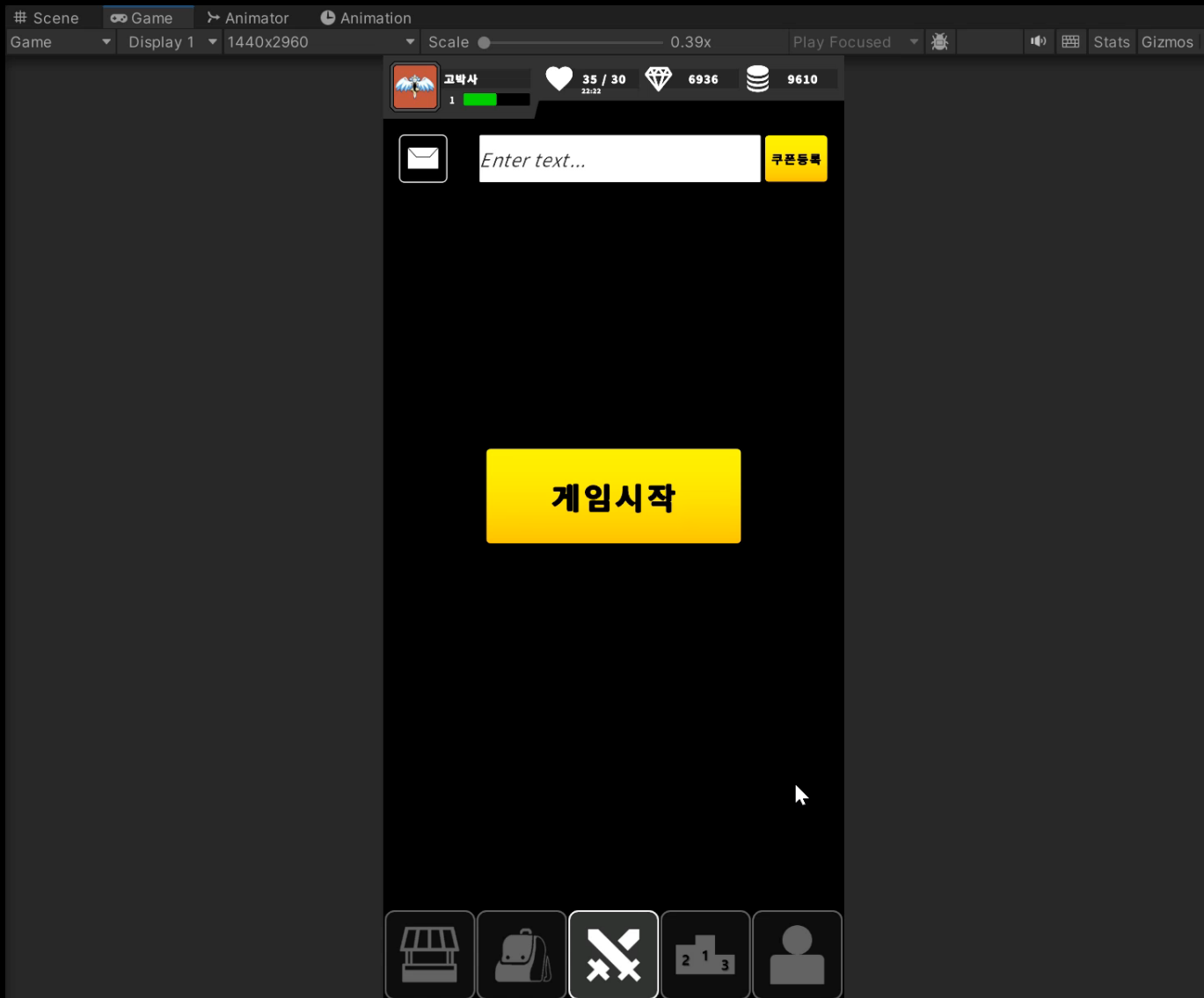
- Runtime Only | FriendReceivedRequest.OnClickA | Friend_Rec

Red boxes highlight the 'Friend_ReceivedRequest' object in the Hierarchy panel, the 'Button' component in the Inspector, and the 'Friend_Rec' object in the event list. A red arrow points from the 'Friend_ReceivedRequest' object to the 'Friend_Rec' object in the event list.



친구 수락 대기 목록

■ 결과 화면





친구 수락 대기 목록

■ 결과 화면 (Backend Console)

Backend Console ProjectA

ProjectA 개발 모드

☆ 유저 관리 게임 유저 생성

대상 상세 검색

번호 유저 번호

<input type="checkbox"/>	1	a06beb80-14	친구	1	친구 목록	우편	우편 목록
<input type="checkbox"/>	2	134a7e10-13e	디바이스 정보	2	디바이스 목록	기타 정보	-
<input type="checkbox"/>	3	01a743a0-13e	유저 차단 기간	-	차단	유저 차단 사유	-
<input type="checkbox"/>	4	d1d9de30-13e	탈퇴 여부	정상	탈퇴 신청일시 (탈퇴 예정일시)	탈퇴 사유	-
<input type="checkbox"/>	5	b9981f10-085	메모	-			
<input type="checkbox"/>	6	aedaa3e0-085					

SDK 문서 콘솔 가이드

10개씩 보기

확인

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



친구 수락 대기 목록

■ 결과 화면 (Backend Console)

The screenshot displays the Backend Console interface for 'ProjectA'. A modal window titled '친구 목록' (Friend List) is open, showing a table of pending friend requests. The table has the following data:

번호	유저 번호	닉네임	수락일
1	01a743a0-13ed-11ee-9ea3-47608d5789df	user04	2023.07.05 15:25

Below the table, there is a pagination control showing '1' and a '10개씩 보기' (View 10) dropdown menu. A '확인' (Confirm) button is located at the bottom right of the modal.

The background interface shows a sidebar with navigation options like '즐거찾기', '대시보드', '서버 설정', '프로젝트', '인증 정보', '푸시', '스토어 정보', '소셜', '예산 알림', '요금제', '데이터베이스', '유저 관리', '유저 접근 관리', '게임 정보 관리', '랭킹 관리', '우편 관리', '푸시 관리', '쿠폰 관리', and '차트 관리'. The main content area shows a list of users with columns for '번호', '유저 번호', '닉네임', and '고박사'.



친구 수락 대기 목록

■ 친구 거절

■ RejectFriend() 메소드를 호출해 친구 거절

□ BackendFriendSystem Script 수정

```
141 Backend.Friend.GetReceivedRequestList(callback =>
142 {
143     if ( !callback.IsSuccess() )...
144
145     // JSON 데이터 파싱 성공
146     try
147     {
148         LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];
149
150         // 받아온 데이터의 개수가 0이면 데이터가 없는 것
151         if ( jsonData.Count <= 0 )...
152
153         // 친구 수락 대기 목록에 있는 모든 UI 비활성화
154         receivedRequestPage.DeactivateAll();
155
156         foreach ( LitJson.JsonData item in jsonData )
157         {
158             FriendData friendData = new FriendData();
159
160             friendData.nickname = item["nickname"].ToString().Equals("True") ? "NONAME" : item["nickname"].ToString();
161             friendData.inDate = item["inDate"].ToString();
162             friendData.createdAt = item["createdAt"].ToString();
163
164             // 친구 요청을 받은지 3일이 지났다면 자동으로 친구 거절
165             if ( IsExpirationDate(friendData.createdAt) )
166             {
167                 RejectFriend(friendData);
168                 continue;
169             }
170
171             // 현재 friendData 정보를 바탕으로 친구 수락 대기 UI 활성화
172             receivedRequestPage.Activate(friendData);
173         }
174     }
175 }
```



친구 수락 대기 목록

- BackendFriendSystem Script 수정 (계속)

```
139 public void GetReceivedRequestList()...
184
185 public void AcceptFriend(FriendData friendData)...
198
199 public void RejectFriend(FriendData friendData)
200 {
201     Backend.Friend.RejectFriend(friendData.inDate, callback =>
202     {
203         if ( !callback.IsSuccess() )
204         {
205             Debug.LogError($"친구 거절 중 에러가 발생했습니다. : {callback}");
206             return;
207         }
208
209         Debug.Log($"{{friendData.nickname}}의 친구 요청을 거절했습니다. : {callback}");
210     });
211 }
212
213 private bool IsExpirationDate(string createdAt)...
248 }
```



친구 수락 대기 목록

- “친구거절” 버튼을 눌렀을 때 호출할 `OnClickRejectRequest()` 메소드 정의
 - `FriendReceivedRequest` Script 수정

```
1 public class FriendReceivedRequest : FriendBase
2 {
3     public override void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData) ...
8
9     public void OnClickAcceptRequest() ...
16
17     public void OnClickRejectRequest()
18     {
19         // 친구 UI 오브젝트 삭제
20         friendPage.Deactivate(gameObject);
21         // 친구 요청 거절 (Backend Console)
22         backendFriendSystem.RejectFriend(friendData);
23     }
24 }
```



친구 수락 대기 목록

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

The screenshot displays the Unity Inspector window for a ButtonReject component. The Hierarchy panel on the left shows the object structure, with Friend_ReceivedRequest selected. The Inspector panel shows the following configuration for the ButtonReject component:

- Tag: Untagged
- Layer: UI
- Rect Transform
- Canvas Renderer
- Image
- Button (selected):
 - 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
 - On Click ():
 - Runtime Only:
 - Event: FriendReceivedRequest.OnClickR
 - Target: Friend_Rec

The Assets > Prefabs panel on the right shows the Friend_ReceivedRequest prefab selected.



친구 수락 대기 목록

■ 결과 화면

The screenshot displays a game interface with a top status bar showing a character named '교박사' (Gyobaksa) with 35/30 hearts, 6936 diamonds, and 9610 coins. The main screen features a text input field with the placeholder 'Enter text...' and a yellow '쿠폰등록' (Coupon Register) button. A large yellow '게임시작' (Start Game) button is centered at the bottom. The console on the right shows several log messages, with the following message highlighted in a red box:


```
[15:30:04] 닉네임의 친구 요청을 거절했습니다. : statusCode : 204  
message : Success
```

Below this message, another log entry is visible: '닉네임의 친구 요청을 거절했습니다. : statusCode : 204 message : Success'. The console also shows a warning message at [15:30:01] and several success messages at [15:29:55].






친구 수락 대기 목록

■ 결과 화면 - 자동 친구 거절



교박사
❤️ 35 / 30
22:22
💎 6936
🏠 9610

친구목록
요청대기
수락대기

	user03	35시간 남음	수락	거절
	NONAME	-14시간 남음	수락	거절


교박사
❤️ 35 / 30
22:22
💎 6936
🏠 9610

친구목록
요청대기
수락대기

	user03	35시간 남음	수락	거절
---	---------------	---------	--	--



< 적용 전 >



< 적용 후 >

친구 목록

- 친구 목록 조회
- 친구 목록 UI 제작
- 친구 목록 UI 출력
- 친구 삭제



친구 목록

■ 친구 목록 조회

■ 친구 목록 조회 메소드

```
GetFriendList(int limit=100, int offset=0);
```

친구 목록 조회

limit : 불러올 친구 목록의 수 (default : 100)

offset : 불러올 친구 기준 숫자

ex) GetFriendList(5); -> 친구 5명 목록 조회 (1-5)

ex) GetFriendList(5, 5); -> 처음 5명 이후의 친구 5명 조회 (6-10)



친구 목록

■ 친구 목록 JsonData

```
{
  "rows":
  [
    // 닉네임이 있는 유저
    {
      "nickname": "고박사"           // 유저의 닉네임
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "createdAt": "2023-06-26T02:22:22.022Z" // 친구가 된 시간
      "lastLogin": "2023-06-26T02:22:22.022Z" // 마지막 접속 시간
    },
    // 닉네임이 없는 유저
    {
      "inDate": "2023-06-26T02:22:22.022Z" // 유저의 inDate
      "createdAt": "2023-06-26T02:22:22.022Z" // 친구가 된 시간
      "lastLogin": "2023-06-26T02:22:22.022Z" // 마지막 접속 시간
    },
  ]
}
```



친구 목록

■ 트랜잭션(Transaction)

트랜잭션(Transaction)이란?

데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

뒤끝의 트랜잭션은 최대 10개의 작업을 하나로 묶어 처리할 수 있는 기능이다.

트랜잭션 쓰기

```
BackendReturnObject TransactionWriteV2(TransactionValue[] transactionList);
```

```
BackendReturnObject TransactionWriteV2(List<TransactionValue> transactionList);
```

- inDate로 검색하는 TransactionValue의 경우, SetUpdate()와 SetDelete() 대신 SetUpdateV2()와 SetDeleteV2()의 작업만 사용할 수 있습니다.
- 작업 리스트에 존재하는 요청들 중 하나 이상의 작업에서 에러가 발생할 경우 모든 요청들은 실패 처리가 됩니다.
- 모든 요청이 성공해야 모든 요청들이 성공 처리가 됩니다.



친구 목록

■ 트랜잭션(Transaction) (계속)

트랜잭션 읽기

TransactionReadV2(TransactionValue[] transactionList);

TransactionReadV2(List<TransactionValue> transactionList);

- inDate를 이용한 데이터 검색의 경우, SetGet() 대신 SetGetV2()의 작업만 사용할 수 있습니다.
- 작업 리스트에 존재하는 요청들 중 하나 이상의 작업에서 에러가 발생할 경우 모든 요청들은 실패 처리가 됩니다.
- 모든 요청이 성공해야 모든 요청들이 성공 처리가 됩니다.
- where 조건으로 다수의 데이터가 검색될 경우, 제일 최근에 삽입된 데이터 1개만 리턴 됩니다.
- 서로 다른 where조건일지라도 검색되는 데이터가 일치한다면 에러가 발생할 수 있습니다.

TransactionValue는 게임 정보 관리에서 트랜잭션 기능을 사용하기 위한 클래스입니다.

TransactionValue의 리스트 혹은 배열을 TransactionRead 혹은 TransactionWrite 메소드에 추가하여 각각의 읽기/쓰기 트랜잭션 작업을 수행할 수 있습니다.

- 테이블 조회 명령 SetGet(Where), SetGetV2(inDate)
- 테이블 삽입 명령 SetInsert()
- 테이블 수정 명령 SetUpdate(Where), SetUpdateV2(inDate)
- 테이블 삭제 명령 SetDelete(Where), SetDeleteV2(inDate)



친구 목록

■ 테이블 조회 명령 SetGet(Where)

```
SetGet(string tableName, Where where);
```

tableName : 조회할 테이블 명

where : 검색할 where 절

조건 없이 모든 데이터 조회

```
new Where()
```

특정 유저의 데이터만 조회

```
Where where = new Where();
```

```
where.Equal("owner_inDate", "검색할 유저의 gamerInDate");
```

게임 정보 관리 테이블의 데이터를 검색할 때 private 테이블의 경우 자기 자신의 데이터만 조회할 수 있기 때문에 다른 유저의 데이터를 검색하려면 테이블을 public으로 설정해야 한다.



친구 목록

- TransactionReadV2()를 이용해 조회한 유저들의 게임 정보 JsonData

```
{
  "Responses":
  [
    {
      "owner_inDate": "2020-02-22T02:22:22.022Z" // 해당 row를 소유한 유저의 inDate
      "client_date": "2020-02-22T02:22:22.022Z" // 해당 row를 insert 한 시간
      "inDate": "2020-02-22T02:22:22.022Z" // 해당 row의 inDate
      "updatedAt": "2020-02-22T02:22:22.022Z" // 해당 row가 마지막으로 업데이트된 시간

      ... // 클라이언트에 저장된 데이터
    },
    {
      ...
    },
    ...
  ]
}
```



친구 목록

- 유저 정보 테이블(USER_DATA)의 분류를 public으로 변경


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

ProjectA ⚠️ 개발 모드

★ 게임 정보 관리 + 테이블 생성 🗑️ 삭제 테이블 분류 - 전체 스키마 - 전체 테이블명으로 검색 🔍 ⚙️ SDK 문서 📄 콘솔 가이드 📄

테이블 데이터

*테이블은 최대 100개까지 생성할 수 있습니다.

<input type="checkbox"/>	테이블명	설명	분류	스키마 정의	상태	
<input type="checkbox"/>	USER_DATA		private	스키마 미정의	활성	

< 1 > 10개씩 보기 ▾

📄 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

회사소개 이용약관 서비스수준협약 개인정보처리방침

© AFI, Inc. All rights reserved.



친구 목록

- 유저 정보 테이블(USER_DATA)의 분류를 public으로 변경 (계속)

The screenshot shows the Backnd Console interface for ProjectA. A modal dialog titled '테이블 수정' (Table Edit) is open, allowing the user to modify the 'USER_DATA' table. The dialog includes the following fields and options:

- 분류 (Category):** Radio buttons for 'Private' and 'Public'. The 'Public' option is selected and highlighted with a red box.
- 테이블명 (Table Name):** Text input field containing 'USER_DATA'.
- 테이블 설명 (Table Description):** Empty text input field.
- 스키마 정의 (Schema Definition):** Radio buttons for '사용' (Used) and '미사용' (Not Used).
- 상태 (Status):** Radio buttons for '활성' (Active) and '비활성' (Inactive).
- Buttons:** '확인' (Confirm) and '취소' (Cancel). The '확인' button is highlighted with a red box.

Background interface details:

- Header: Backnd Console, ProjectA, settings icon, notification bell, help icon, user profile icon.
- Navigation: 게임 정보 관리, 테이블 생성, 삭제, 테이블 분류 - 전체, 스키마 - 전체, 테이블명으로 검색, SDK 문서, 콘솔 가이드.
- Table List: 테이블, 데이터, *테이블은 최대 100개까지 생성할 수 있습니다., 테이블명, USER_DATA, 상태, 활성, 10개씩 보기.
- Left Sidebar: 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.
- Footer: 회사소개, 이용약관, 서비스수준협약, 개인정보처리방침, © AFI, Inc. All rights reserved.



친구 목록

- 마지막 접속 날짜, 레벨 변수 선언
 - FriendData Script 수정

```
1 public class FriendData
2 {
3     public string nickname; // 닉네임
4     public string inDate; // 해당 유저의 inDate
5     public string createdAt; // 친구 요청 보낸 시간 / 친구 요청 받은 시간 / 친구가 된 시간
6     public string lastLogin; // 마지막 접속 날짜
7
8     // level과 같이 출력하고 싶은 유저정보가 있으면 추가
9     public string level; // 해당 유저의 level
10
11 public override string ToString()
12 {
13     string result = string.Empty;
14     result += $"닉네임 : {nickname}\n";
15     result += $"inDate : {inDate}\n";
16     result += $"createdAt : {createdAt}\n";
17     result += $"lastLogin : {lastLogin}\n";
18     result += $"level : {level}\n";
19
20     return result;
21 }
22 }
```



친구 목록

- GetFriendList() 메소드를 호출해 친구 목록 조회
 - BackendFriendSystem Script 수정

```
1  using System.Collections.Generic;
2  using UnityEngine;
3  using System;
4  using BackEnd;
5
6  public class BackendFriendSystem : MonoBehaviour
7  {
8      [SerializeField]
9      private FriendSentRequestPage    sentRequestPage;
10     [SerializeField]
11     private FriendReceivedRequestPage receivedRequestPage;
12
13     private string GetUserInfoBy(string nickname)...
50
51     public void SendRequestFriend(string nickname)...
71
72     public void GetSentRequestList()...
125
126     public void RevokeSentRequest(string inDate)...
139
140     public void GetReceivedRequestList()...
192
193     public void AcceptFriend(FriendData friendData)...
206
207     public void RejectFriend(FriendData friendData)...
220
```



친구 목록

BackendFriendSystem Script 수정 (계속)

```
221 public void GetFriendList()
222 {
223     Backend.Friend.GetFriendList(callback =>
224     {
225         if ( !callback.IsSuccess() )
226         {
227             Debug.LogError($"친구 목록 조회 도중 에러가 발생했습니다. : {callback}");
228             return;
229         }
230
231         // JSON 데이터 파싱 성공
232         try
233         {
234             LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];
235
236             // 받은 데이터의 개수가 0이면 데이터가 없는 것
237             if ( jsonData.Count <= 0 )
238             {
239                 Debug.LogWarning("친구 목록 데이터가 없습니다.");
240                 return;
241             }
242         }
```



친구 목록

□ BackendFriendSystem Script 수정 (계속)

```
243 List<TransactionValue> transactionList = new List<TransactionValue>();
244 List<FriendData> friendDataList = new List<FriendData>();
245
246 foreach ( LitJson.JsonData item in jsonData )
247 {
248     FriendData friendData = new FriendData();
249
250     friendData.nickname = item["nickname"].ToString().Equals("True") ? "NONAME" : item["nickname"].ToString();
251     friendData.inDate = item["inDate"].ToString();
252     friendData.createdAt = item["createdAt"].ToString();
253     friendData.lastLogin = item["lastLogin"].ToString();
254
255     friendDataList.Add(friendData);
256
257     // friendData.inDate를 가지는 친구의 UserGameData 정보 불러오기
258     Where where = new Where();
259     where.Equal("owner_inDate", friendData.inDate);
260     transactionList.Add(TransactionValue.SetGet(Constants.USER_DATA_TABLE, where));
261 }
262
```




친구 목록

BackendFriendSystem Script 수정 (계속)

```
263 Backend.GameData.TransactionReadV2(transactionList, callback =>
264 {
265     if ( !callback.IsSuccess() )
266     {
267         Debug.LogError($"Transaction Error : {callback}");
268         return;
269     }
270
271     LitJson.JsonData userData = callback.GetFlattenJSON()["Responses"];
272
273     if ( userData.Count <= 0 )
274     {
275         Debug.LogWarning($"데이터가 존재하지 않습니다.");
276         return;
277     }
278
279     for ( int i = 0; i < userData.Count; ++ i )
280     {
281         // 친구 레벨 정보 설정
282         friendDataList[i].level = $"Lv. {userData[i]["level"]}";
283     }
284     });
285 }
286 // JSON 데이터 파싱 실패
287 catch ( Exception e )
288 {
289     // try-catch 에러 출력
290     Debug.LogError(e);
291 }
292 });
293 }
294
295 private bool IsExpirationDate(string createdAt)...
```

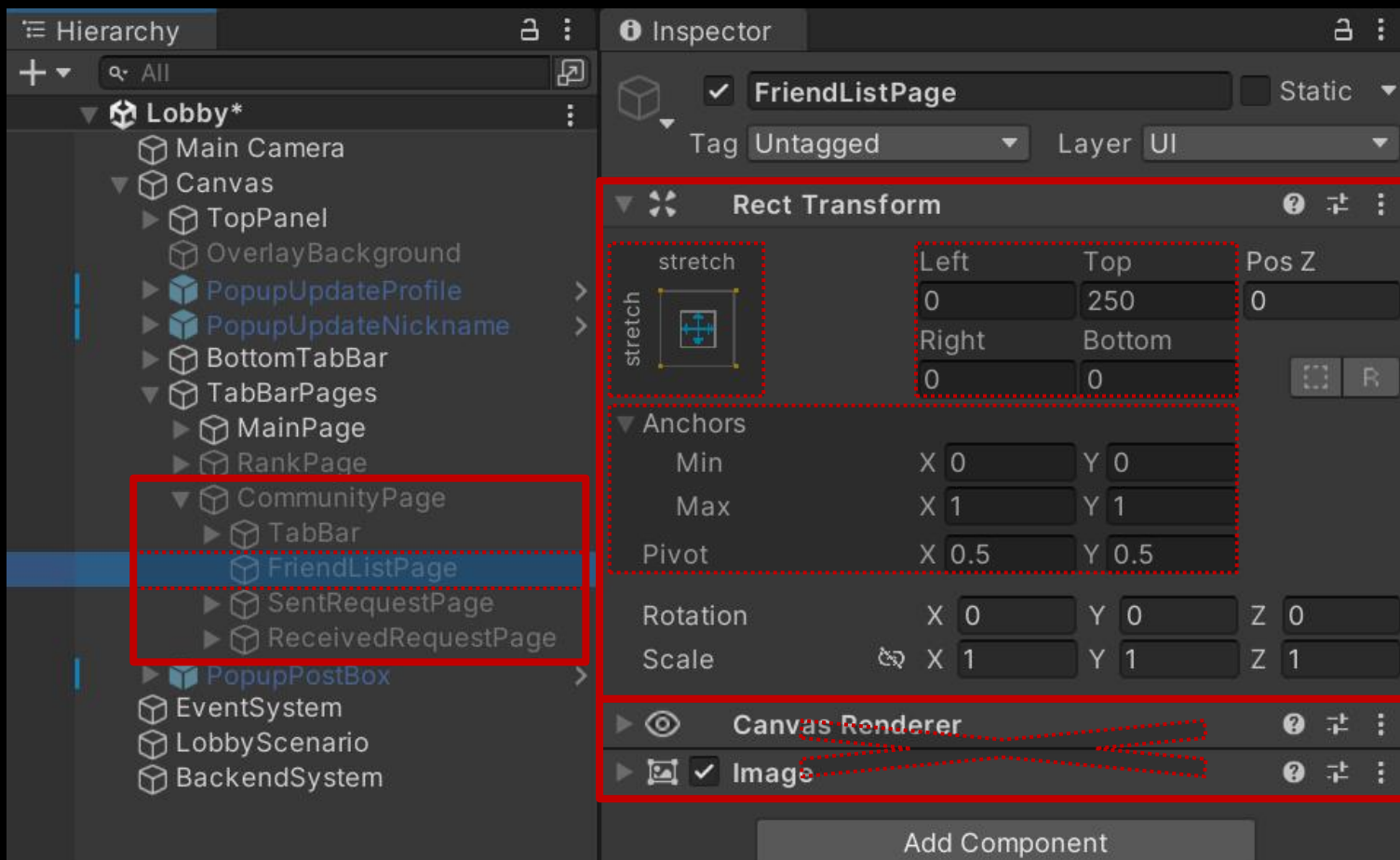


친구 목록

■ 친구 목록 UI 제작

■ 친구 페이지를 관리하는 Panel UI 생성 및 설정

□ GameObject - UI - Panel





친구 목록

- 친구 목록을 출력하는 Scroll View UI 생성 및 설정
 - GameObject - UI - Scroll View

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby*' with 'FriendListPage' selected, containing 'FriendList', 'Viewport', 'Content', 'Scrollbar Horizontal', and 'Scrollbar Vertical'. The Inspector panel on the right shows the properties for the selected 'FriendList' object.

Inspector Panel Details:

- FriendList** (GameObject): Tag: Untagged, Layer: UI.
- Rect Transform**:
 - stretch: stretch (checked), stretch (checked)
 - Left: 0, Top: 0, Right: 0, Bottom: 0, Pos Z: 0
 - 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
- Canvas Renderer**
- Image**
- Scroll Rect**: Content: Content (Rect Transform), Horizontal: (unchecked), Vertical: (checked)



친구 목록

- Scrollbar가 보이지 않도록 Width를 0으로 설정

The image shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Scrollbar Vertical' component is selected under the 'Lobby*' hierarchy. The Inspector panel shows the 'Rect Transform' component with the following settings:

Property	Value
Pos X	0
Top	0
Pos Z	0
Width	0
Bottom	17
Min X	1
Min Y	0
Max X	1
Max Y	1
Pivot X	1
Pivot Y	1
Rotation X	0
Rotation Y	0
Rotation Z	0
Scale X	1
Scale Y	1
Scale Z	1

The 'Width' field is highlighted with a red dashed box, indicating it is set to 0. The 'Scrollbar' component is also visible in the Inspector panel below the 'Rect Transform' component.



친구 목록

- Content 오브젝트에 컴포넌트 추가 및 설정

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree view of the scene hierarchy. The 'Content' object is selected and highlighted with a red box. The Inspector panel on the right shows the properties of the selected 'Content' object. The 'Content' component is checked and has 'Tag' set to 'Untagged' and 'Layer' set to 'UI'. Below it, the 'Rect Transform' component is visible. The 'Vertical Layout Group' component is expanded and highlighted with a red dashed box. It has the following settings: 'Padding' (Left: 20, Right: 20, Top: 20, Bottom: 20), 'Spacing' (20), 'Child Alignment' (Upper Center), 'Reverse Arrangement' (unchecked), 'Control Child Size' (Width checked, Height unchecked), 'Use Child Scale' (Width unchecked, Height unchecked), and 'Child Force Expand' (Width checked, Height checked). Below the 'Vertical Layout Group' is the 'Content Size Fitter' component, which is also expanded and highlighted with a red dashed box. It has 'Horizontal Fit' set to 'Unconstrained' and 'Vertical Fit' set to 'Preferred Size'. At the bottom of the Inspector panel, there is an 'Add Component' button.



친구 목록

- “[친구 목록..” 텍스트를 출력하는 “Text - TextMeshPro” UI 생성 및 설정
 - GameObject - UI - “Text - TextMeshPro”

Hierarchy: Lobby* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > MainPage > RankPage > CommunityPage > TabBar > FriendListPage > FriendList > TextSystem

Inspector: TextSystem (Tag: Untagged, Layer: UI)

Rect Transform:

stretch	Left	Pos Y	Pos Z
	0	0	0
middle	Right	Height	
	0	50	

Anchors:

Min	X	0	Y	0.5
Max	X	1	Y	0.5
Pivot	X	0.5	Y	0.5

Rotation: X 0, Y 0, Z 0

Scale: X 1, Y 1, Z 1

Text Input: [친구 목록]이 비어있습니다.

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: 60
- Auto Size:

Alignment: Left Center Right Justified Full Justified

Wrapping: Enabled

Overflow: Overflow

Horizontal Mapping: Character

Vertical Mapping: Character



친구 목록

- FriendList 오브젝트의 "Toggle" 컴포넌트 OnValueChanged() 이벤트 등록

The screenshot displays the Unity Inspector window for a **FriendList** object. The **Toggle** component is selected, and its properties are visible. The **On Value Changed (Boolean)** event is configured to call **GameObject.SetActive** on the **FriendList** object.

Inspector Settings:

- Interactable:**
- Transition:** Color Tint
 - Target Graphic: Background (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
- Is On:**
- Toggle Transition:** Fade
- Graphic:** Checkmark (Image)
- Group:** TabBar (Toggle Group)
- On Value Changed (Boolean):**
 - Runtime Only
 - GameObject.SetActive
 - FriendList



친구 목록

- SentRequestList 오브젝트의 "Toggle" 컴포넌트 IsOn 변수 설정

The image shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **SentRequestList** object is selected and highlighted with a red box. The Inspector panel shows the **Toggle** component selected, with its properties visible. The **Is On** property is highlighted with a red dashed box, and its value is set to **False**. The **On Value Changed (Boolean)** event is configured to call **GameObject.SetActive** on the **SentRequestList** object.

Hierarchy Panel:

- Lobby*
 - Main Camera
 - Canvas
 - TopPanel
 - OverlayBackground
 - PopupUpdateProfile
 - PopupUpdateNickname
 - BottomTabBar
 - TabBarPages
 - MainPage
 - RankPage
 - CommunityPage
 - TabBar
 - FriendList
 - SentRequestList**
 - ReceivedRequestList
 - FriendListPage
 - SentRequestPage
 - ReceivedRequestPage
 - PopupPostBox
 - EventSystem
 - LobbyScenario
 - BackendSystem

Inspector Panel:

- SentRequestList** (Static)
 - Tag: Untagged, Layer: UI
 - Rect Transform**
 - Toggle**
 - Interactive:
 - Transition: Color Tint
 - Target Graphic: Background (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
 - Is On**:
 - Toggle Transition: Fade
 - Graphic: Checkmark (Image)
 - Group: TabBar (Toggle Group)
 - On Value Changed (Boolean): Runtime Only, GameObject.SetActive, SentRequestList



친구 목록

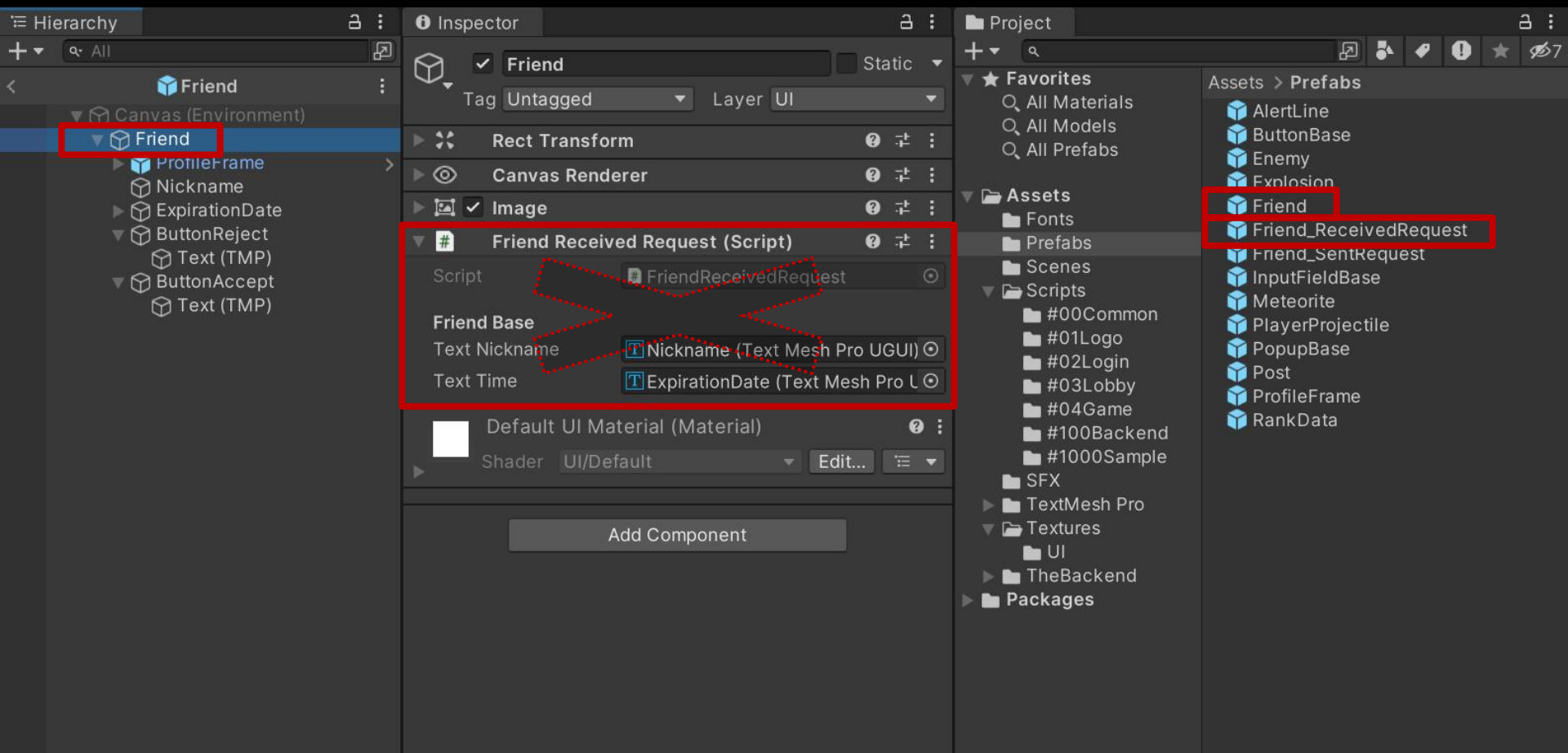
■ SentRequestPage 오브젝트 비활성화

The image shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **SentRequestPage** object is selected and highlighted with a red box. In the Inspector panel, the **SentRequestPage** object is selected, and its **Friend Sent Request Page (Script)** component is active. The **Friend Sent Request Page (Script)** component is checked, and its **Script** is set to **FriendSentRequestPage**. The **Common** section shows **Backend Friend System** set to **BackendSystem (Backend Friend)**. The **Friend Page Base** section shows **Friend Prefab** set to **Friend_SentRequest**, **Parent Content** set to **Content (Rect Transform)**, and **Text System** set to **TextSystem**. The **Send Request Friend** section shows **Input Field Nickname** set to **InputFieldBase (TMP_Input Field)** and **Text Result** set to **TextResult (Fade Effect_TMP)**. The **Add Component** button is visible at the bottom.



친구 목록

- 친구 UI를 관리하는 오브젝트 생성 및 설정
 - Friend_ReceivedRequest 프리팹을 Ctrl+D로 복제





친구 목록

□ 닉네임 Text UI 설정

The image shows the Unity Inspector window for a 'Nickname' UI element. The Hierarchy panel on the left shows the object's path: Friend > Canvas (Environment) > Friend > ProfileFrame > Nickname. The Inspector panel on the right shows the following settings:

- Component:** Nickname (checked), Static (unchecked)
- Tag:** Untagged
- Layer:** UI
- Rect Transform:** (highlighted with a red box)
 - Position:** Pos X: 180, Pos Y: -10, Pos Z: 0
 - Size:** Width: 400, Height: 75
 - Anchor:** Min (X: 0, Y: 1), Max (X: 0, Y: 1), Pivot (X: 0, Y: 1)
 - Rotation:** X: 0, Y: 0, Z: 0
 - Scale:** X: 1, Y: 1, Z: 1
- Canvas Renderer:** (checked)
- TextMeshPro - Text (UI):** (checked)
- Material:** NotoSansKR-Bold SDF Material (Material)
- Shader:** TextMeshPro/Distance Fi

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



친구 목록

레벨 Text UI 설정

Hierarchy: Friend > ProfileFrame > Nickname > Level

Inspector: Level (Tag: Untagged, Layer: UI)

Property	Value
Pos X	180
Pos Y	10
Width	400
Height	75

Anchor: bottom-left

Rotation: X 0, Y 0, Z 0

Scale: X 1, Y 1, Z 1

Canvas Renderer

TextMeshPro - Text (UI)

NotoSansKR-Bold SDF Material (Material)

Shader: TextMeshPro/Distance Fi

Add Component

Nickname 오브젝트를 Ctrl+D로 복제한 후 이름을 Level로 변경

Text Input: Lv. 1

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: 36

Auto Size: []

Vertex Color: [Color Picker]

Color Gradient: [Color Gradient]

Override Tags: []

Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0

Alignment: [Left] [Center] [Right] [Justify] [Full]

Wrapping: Enabled

Overflow: Overflow

Horizontal Mapping: Character

Vertical Mapping: Character



친구 목록

- 마지막 접속 날짜 Text UI 설정

ExpirationDate 오브젝트의 이름을 Date로 변경

Text Input Enable RTL Editor

마지막 접속 날짜

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 36

Auto Size

Vertex Color

Color Gradient

Override Tags

Spacing Options (em) Character 0 Word 0 Line 0 Paragraph 0

Alignment

Wrapping Enabled

Overflow Overflow

Horizontal Mapping Character

Vertical Mapping Character



친구 목록

□ "친구삭제" Button UI 설정

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure: Friend (Canvas Environment) > Friend > ButtonDeleteFriend. The Inspector panel on the right shows the properties of the selected ButtonDeleteFriend component.

Hierarchy Panel:

- Friend (Canvas Environment)
 - Friend
 - ButtonReject 오브젝트의 이름을 ButtonDeleteFriend로 변경
 - Date
 - ButtonDeleteFriend** (highlighted with a red box)
 - Text (TMP)
 - ButtonAccept

Inspector Panel:

- ButtonDeleteFriend** (Static: unchecked, Tag: Untagged, Layer: UI)
- Rect Transform** (highlighted with a red box)
 - right, middle (pivot points)
 - Pos X: -10, Pos Y: 0, Pos Z: 0
 - Width: 150, Height: 150
 - Anchors
 - Min: X 1, Y 0.5
 - Max: X 1, Y 0.5
 - Pivot: X 1, Y 0.5
 - Rotation: X 0, Y 0, Z 0
 - Scale: X 1, Y 1, Z 1
- Canvas Renderer
- Image (checked)
- Button (checked)
- Default UI Material (Material)
 - Shader: UI/Default

Annotations:

- ButtonReject 오브젝트의 이름을 ButtonDeleteFriend로 변경
- ButtonAccept 오브젝트 삭제

Buttons: Add Component



친구 목록

- "친구삭제" Button UI 설정 (계속)

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Friend' with 'Text (TMP)' selected and highlighted by a red box. The Inspector panel on the right shows the properties for the selected 'Text (TMP)' object. The 'TextMeshPro - Text (UI)' component is expanded, showing the text '친구 삭제' in a dashed red box. Below the text, the 'Main Settings' section is visible, including 'Font Asset' set to 'NotoSansKR-Bold SDF (TMP_Fc)', 'Font Style' with 'B' (Bold) selected, and 'Font Size' set to 40.

Hierarchy Panel:

- Friend
 - Canvas (Environment)
 - Friend
 - ProfileFrame
 - Nickname
 - Level
 - Date
 - ButtonDeleteFriend
 - Text (TMP)**

Inspector Panel:

- Text (TMP) [Static]
- Tag: Untagged, Layer: UI
- Rect Transform
- Canvas Renderer
- TextMeshPro - Text (UI)**
 - Text Input: 친구 삭제 (Enable RTL Editor:)
 - 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:)
 - Vertex Color: [Color Picker]
 - Color Gradient: [Color Gradient]



친구 목록

■ 친구 목록 UI 출력

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

```
1 public class FriendPage : FriendPageBase
2 {
3     private void OnEnable()
4     {
5         // [친구] 목록 불러오기
6         backendFriendSystem.GetFriendList();
7     }
8
9     private void OnDisable()
10    {
11        DeactivateAll();
12    }
13 }
```




친구 목록

- FriendListPage 오브젝트의 "FriendPage" 컴포넌트 변수 설정

The screenshot displays the Unity Inspector window for the **FriendListPage** object. The **Friend Page (Script)** component is selected, and its variables are being configured:

- Backend Friend System:** Set to **BackendSystem (Backend Friend)**.
- Friend Prefab:** Set to **Friend**.
- Parent Content:** Set to **Content (Rect Transform)**.
- Text System:** Set to **TextSystem**.

Red boxes and arrows highlight the following elements in the Hierarchy and Project panels:

- FriendListPage** in the Hierarchy panel.
- Content** and **TextSystem** under **FriendListPage** in the Hierarchy panel.
- BackendSystem** under **Assets > Prefabs** in the Project panel.
- Friend** under **Assets > Prefabs** in the Project panel.



친구 목록

- 자식 클래스에서 textTime에 접근할 수 있도록 접근 지정자 설정
 - FriendBase Script 수정

```
1 using UnityEngine;
2 using TMPro;
3 using BackEnd;
4 using System;
5
6 public class FriendBase : MonoBehaviour
7 {
8     [Header("Friend Base")]
9     [SerializeField]
10    private TextMeshProUGUI textNickname; // 닉네임
11    [SerializeField]
12    protected TextMeshProUGUI textTime; // 만료시간, 접속시간 등의 시간 정보
13
14    protected BackendFriendSystem backendFriendSystem;
15    protected FriendPageBase friendPage;
16    protected FriendData friendData;
17
18    public virtual void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData) ...
19
20
21
22
23
24
25
26
27    public void SetExpirationDate() ...
28
29 }
```



친구 목록

- 친구 UI를 제어하는 스크립트 생성 및 작성
 - C# Script 생성 후 스크립트의 이름을 "Friend"로 변경

```
1 using UnityEngine;
2 using TMPro;
3
4 public class Friend : FriendBase
5 {
6     [SerializeField]
7     private TextMeshProUGUI textLevel;
8
9     public override void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData)
10    {
11        base.Setup(friendSystem, friendPage, friendData);
12
13        textLevel.text = friendData.level;
14        textTime.text = System.DateTime.Parse(friendData.lastLogin).ToString();
15    }
16 }
```



친구 목록

- Friend 프리팹에 "Friend" 컴포넌트 추가 및 설정

The screenshot displays the Unity Inspector window for a 'Friend' component. The 'Friend (Script)' component is selected, and its 'Friend Base' section is expanded. Three red dashed boxes highlight the 'Nickname (Text Mesh Pro UGUI)', 'Date (Text Mesh Pro UGUI)', and 'Level (Text Mesh Pro UGUI)' fields. Red arrows point from the 'Nickname', 'Level', and 'Date' entries in the Hierarchy panel to their respective fields in the Inspector. The 'Assets > Prefabs' panel on the right shows the 'Friend' prefab selected.

Hierarchy Panel:

- Canvas (Environment)
 - Friend
 - ProfileFrame
 - Nickname
 - Level
 - Date
 - ButtonDeleteFriend

Inspector Panel:

- Friend
 - Tag: Untagged
 - Layer: UI
 - Rect Transform
 - Canvas Renderer
 - Image
 - Friend (Script)
 - Script: Friend
 - Friend Base
 - Text Nickname: Nickname (Text Mesh Pro UGUI)
 - Text Time: Date (Text Mesh Pro UGUI)
 - Text Level: Level (Text Mesh Pro UGUI)
 - Default UI Material (Material)
 - Shader: UI/Default

Assets > Prefabs Panel:

- AlertLine
- ButtonBase
- Enemy
- Explosion
- Friend
- Friend_ReceivedRequest
- Friend_SentRequest
- InputFieldBase
- Meteorite
- PlayerProjectile
- PopupBase
- Post
- ProfileFrame
- RankData



친구 목록

- friendPage.Activate() 메소드를 호출해 친구 목록 UI 출력
 - BackendFriendSystem Script 수정

```
1  +using ...
5
6  -public class BackendFriendSystem : MonoBehaviour
7  {
8      [SerializeField]
9      private FriendSentRequestPage    sentRequestPage;
10     [SerializeField]
11     private FriendReceivedRequestPage receivedRequestPage;
12     [SerializeField]
13     private FriendPage                friendPage;
14
15     +private string GetUserInfoBy(string nickname)...
52
53     +public void SendRequestFriend(string nickname)...
73
74     +public void GetSentRequestList()...
127
128    +public void RevokeSentRequest(string inDate)...
141
142    +public void GetReceivedRequestList()...
194
195    +public void AcceptFriend(FriendData friendData)...
208
209    +public void RejectFriend(FriendData friendData)...
222
```




친구 목록

BackendFriendSystem Script 수정 (계속)

```
223 public void GetFriendList()
224 {
225     Backend.Friend.GetFriendList(callback =>
226     {
227         if ( !callback.IsSuccess() ) ...
232
233         // JSON 데이터 파싱 성공
234         try
235         {
236             LitJson.JsonData jsonData = callback.GetFlattenJSON()["rows"];
237
238             // 받은 데이터의 개수가 0이면 데이터가 없는 것
239             if ( jsonData.Count <= 0 ) ...
244
245             // 친구 목록에 있는 모든 UI 비활성화
246             friendPage.DeactivateAll();
247
248             List<TransactionValue> transactionList = new List<TransactionValue>();
249             List<FriendData> friendDataList = new List<FriendData>();
250
251             foreach ( LitJson.JsonData item in jsonData ) ...
267
```



친구 목록

BackendFriendSystem Script 수정 (계속)

```
268 Backend.GameData.TransactionReadV2(transactionList, callback =>
269 {
270     if ( !callback.IsSuccess() )...
275
276     LitJson.JsonData userData = callback.GetFlattenJSON()["Responses"];
277
278     if ( userData.Count <= 0 )...
283
284     for ( int i = 0; i < userData.Count; ++ i )
285     {
286         // 친구 레벨 정보 설정
287         friendDataList[i].level = $"Lv. {userData[i]["level"]}";
288         // 현재 friendDataList[i] 정보를 바탕으로 친구 UI 활성화
289         friendPage.Activate(friendDataList[i]);
290     }
291     });
292 }
293 // JSON 데이터 파싱 실패
294 catch ( Exception e )
295 {
296     // try-catch 에러 출력
297     Debug.LogError(e);
298 }
299 });
300 }
301
302 private bool IsExpirationDate(string createdAt)...
337 }
```




친구 목록

- BackendSystem 오브젝트의 "BackendFriendSystem" 컴포넌트 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **BackendSystem** object is selected and highlighted with a red box. In the Inspector panel, the **BackendFriendSystem (Script)** component is selected and highlighted with a red box. The component's variables are listed as follows:

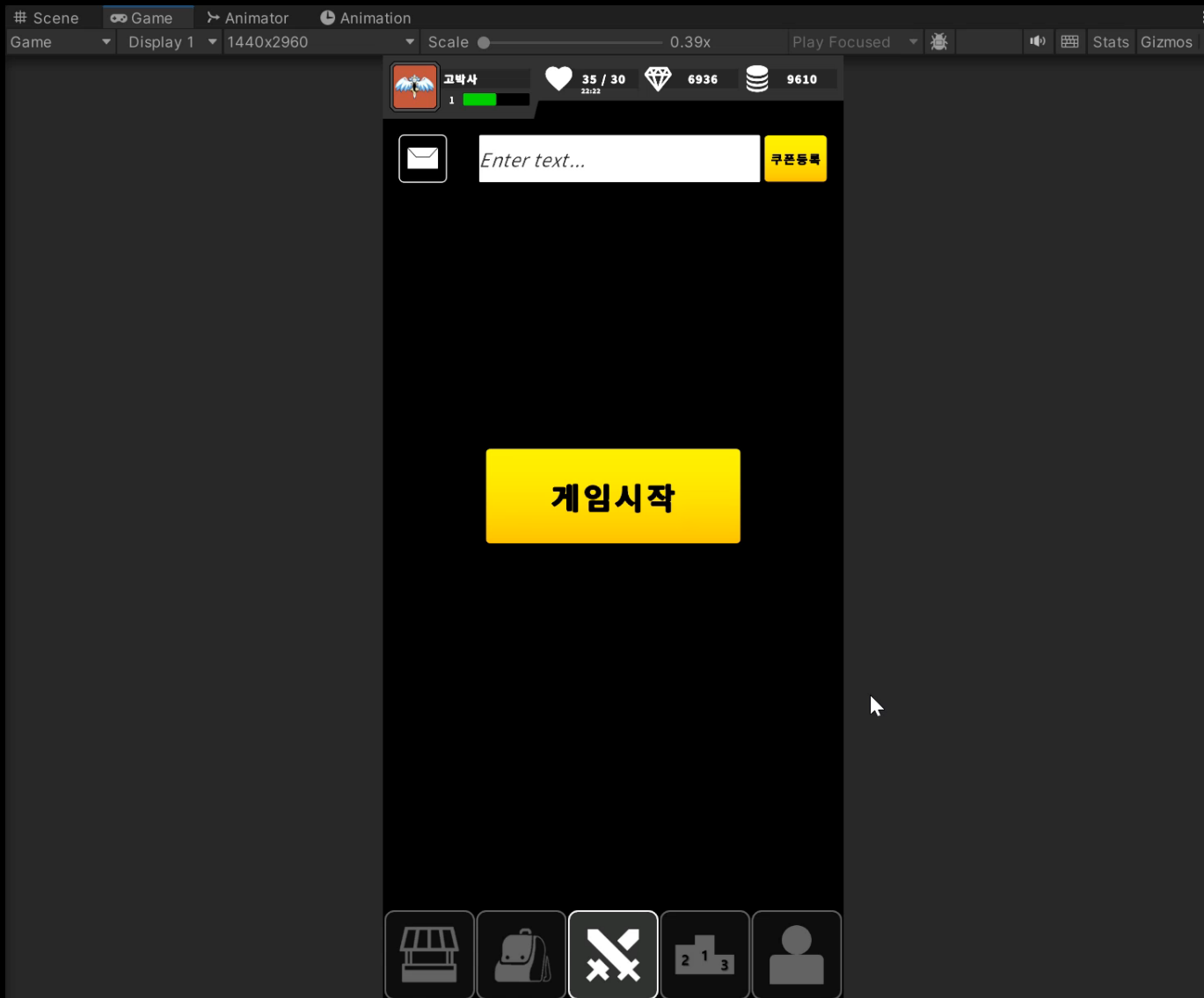
Property	Value
Script	BackendFriendSystem
Sent Request Page	SentRequestPage (Friend Sent F
Received Request Page	ReceivedRequestPage (Friend R
Friend Page	FriendListPage (Friend Page)

Red boxes highlight the **BackendSystem** object in the Hierarchy and the **BackendFriendSystem (Script)** component in the Inspector. A red arrow points from the **FriendListPage (Friend Page)** value in the Inspector to the **FriendListPage** object in the Hierarchy.



친구 목록

■ 결과 화면





친구 목록

■ 친구 삭제

- BreakFriend() 메소드를 호출해 친구 삭제
 - BackendFriendSystem Script 수정

```
195 public void AcceptFriend(FriendData friendData) ...
208
209 public void RejectFriend(FriendData friendData) ...
222
223 public void GetFriendList() ...
301
302 public void BreakFriend(FriendData friendData)
303 {
304     Backend.Friend.BreakFriend(friendData.inDate, callback =>
305     {
306         if ( !callback.IsSuccess() )
307         {
308             Debug.LogError($"친구 삭제 중 에러가 발생했습니다. : {callback}");
309             return;
310         }
311
312         Debug.Log($"{friendData.nickname}와(과) 친구가 해제되었습니다. : {callback}");
313     });
314 }
315
316 private bool IsExpirationDate(string createdAt) ...
351 }
```



친구 목록

- “친구삭제” 버튼을 눌렀을 때 호출할 `OnClickDeleteFriend()` 메소드 정의
 - Friend Script 수정

```
1 using UnityEngine;
2 using TMPro;
3
4 public class Friend : FriendBase
5 {
6     [SerializeField]
7     private TextMeshProUGUI textLevel;
8
9     public override void Setup(BackendFriendSystem friendSystem, FriendPageBase friendPage, FriendData friendData) ...
16
17     public void OnClickDeleteFriend()
18     {
19         // 친구 UI 오브젝트 삭제
20         friendPage.Deactivate(gameObject);
21         // 친구 삭제 (Backend Console)
22         backendFriendSystem.BreakFriend(friendData);
23     }
24 }
```



친구 목록

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

The screenshot displays the Unity Inspector for the **ButtonDeleteFriend** object. The **Button** component is selected, and its **On Click ()** event is configured. The event is set to **Runtime Only** and **Friend.OnClickDeleteFriend**. The target object is **Friend (Fri...)**. The **Friend** object is also visible in the Hierarchy panel, and the **Friend** prefab is visible in the Assets > Prefabs panel.



친구 목록

■ 결과 화면

The screenshot displays a game interface with a dark background. At the top, there's a header with "# Scene", "Game", "Animator", and "Animation" tabs. Below this, there are settings for "Game", "Display 1", "1440x2960", "Scale 0.39x", and "Play Focused".

The main interface features a player profile for "고박사" (Go-bak-sa) with a level indicator "1" and a green progress bar. To the right, there are icons for health (35 / 30), diamonds (6936), and coins (9610). Below the profile is an input field labeled "Enter text..." and a yellow button labeled "쿠폰등록" (Coupon Register).

In the center, there is a large yellow button labeled "게임시작" (Game Start).

At the bottom, there are five icons: a shop, a backpack, crossed swords, a podium with numbers 2, 1, 3, and a person icon.

On the right side, there is a "Console" window showing a list of log messages. The messages include:

- UnityEngine.Debug:Log (object)
- [13:58:51] Level : 8, Max Exp : 12800, Reward Gold : 8888
- UnityEngine.Debug:Log (object)
- [13:58:51] Level : 9, Max Exp : 25600, Reward Gold : 9999
- UnityEngine.Debug:Log (object)
- [13:58:51] Level : 10, Max Exp : 51200, Reward Gold : 100000
- UnityEngine.Debug:Log (object)
- [13:58:51] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200 message : Success
- [13:58:57] user03와(과) 친구가 해제되었습니다. : statusCode : 204 message : Success

The last message is highlighted with a red border.



친구 목록

■ 결과 화면 (Backend Console)

Backend Console ProjectA

ProjectA 개발 모드

★ 즐겨찾기

대시보드

서버 설정

프로젝트

인증 정보

푸시

스토어 정보

소셜

예산 알림

요금제

데이터베이스

유저 관리

유저 접근 관리

게임 정보 관리

랭킹 관리

우편 관리

푸시 관리

쿠폰 관리

차트 관리

회사소개 이용약관 서비스수준협약 개인정보처리방침

유저 관리 게임 유저 생성

대상

상세 검색

번호 유저 번호

1 a06beb80-14

2 134a7e10-13e

3 01a743a0-13e

4 d1d9de30-13e

5 b9981f10-085

6 aedaa3e0-085

aedaa3e0-0859-11ee-8b11-73576fc879f3 수정

유저 아이디	user01	닉네임	고박사
가입일	2023.06.11 22:12	최종 접속일	2023.07.05 22:52
가입 유형	customSignUp	TBC	-
접속 OS	Windows 10 ྪ.0.19045) 64bit	접속 기기	MacBookPro16,1 (Apple Inc.)
길드	-	국가	-
친구	0	친구 목록	우편 목록
디바이스 정보	2	디바이스 목록	기타 정보
유저 차단 기간	-	차단	유저 차단 사유
탈퇴 여부	정상	탈퇴 사유	-
탈퇴 신청일시 (탈퇴 예정일시)	-	탈퇴 사유	-
메모	-		

확인

SDK 문서 콘솔 가이드

중 접속일 국가 접속 OS

2023.06.26 20:36 - Windows 10 ྪ...

2023.06.27 14:31 - Windows 10 ྪ...

2023.07.05 15:14 - Windows 10 ྪ...

2023.06.30 14:04 - Windows 10 ྪ...

2023.07.05 15:13 - Windows 10 ྪ...

2023.07.05 22:52 - Windows 10 ྪ...

10개씩 보기

© AFI, Inc. All rights reserved.