

# Unity Backend

“게임 정보 관리” 기능을 이용한 유저 정보 추가 및 불러오기

Created in 2023-02-13  
Last Updated 2023-03-20  
Unity Version 2022.2.2f1

## *Index*

- ◆ 게임 정보 추가
- ◆ 게임 정보 불러오기

# 게임 정보 추가

- 뒤끝 콘솔에 유저 정보 테이블 생성
- 유니티에 유저 정보 클래스 생성
- 계정을 생성할 때 테이블에 유저 정보 추가



# 게임 정보 추가

## ■ 뒤끝 콘솔에 유저 정보 테이블 생성

### ■ 유저 정보 테이블 생성

The screenshot shows the Backnd Console interface for 'ProjectA'. The left sidebar contains a navigation menu with '뒤끝베이스' (Backbase) and '게임 정보 관리' (Game Information Management) highlighted with red boxes. The main content area shows the 'Game Information Management' page with a '테이블 생성' (Create Table) button highlighted with a red box. Below the table creation area, a black box contains the following instructions:

1. 뒤끝 로그인 후 뒤끝 콘솔 페이지로 이동
2. 뒤끝베이스 - 게임 정보 관리 탭으로 이동
3. "테이블 생성" 버튼을 눌러 새로운 테이블 생성

At the bottom of the page, there is a footer with the text: © AFI, Inc. All rights reserved.



# 게임 정보 추가

## ■ 유저 정보 테이블 생성 (계속)

Backnd Console ProjectA

ProjectA 개발 모드

게임 정보 관리 테이블 생성 삭제

전체 테이블명으로 검색 SDK 문서 콘솔 가이드

테이블 데이터

\*테이블은 최대 100개까지 생성할 수 있습니다.

테이블명

테이블이 없습니다.

상태

### 테이블 생성

테이블은 최대 100개까지 생성할 수 있습니다.

분류  Private  Public

테이블명

테이블 설명

스키마 정의  사용  미사용

상태  활성화  비활성

회사소개 이용약관 서비스수준협약 개인정보처리방침

© AFI, Inc. All rights reserved.



# 게임 정보 추가

## ■ 유저 정보 테이블 생성 (계속)

Backnd Console << ProjectA ⚙️ 🔍 📄 ? 0 관리자 👤

ProjectA ⚠️ 개발 모드

☆ 즐겨찾기

📊 대시보드

서버 설정

🖥️ 프로젝트

🛡️ 인증 정보

🔔 푸시

📁 스토어 정보

👤 소셜

📅 예산 알림

🔗 요금제

🗄️ 뒤글베이스

👤 유저 관리

🕒 유저 접근 관리

📁 게임 정보 관리

🏆 랭킹 관리

✉️ 우편 관리

🔔 푸시 관리

🎫 쿠폰 관리

📊 차트 관리

회사소개 이용약관 서비스수준협약 개인정보처리방침

★ 게임 정보 관리 + 테이블 생성 🗑️ 삭제

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"/>	<b>USER_DATA</b>		private	스키마 미정의	활성 <a href="#">🔗</a>

< 1 >

10개씩 보기 ▾

**방금 생성한 "USER\_DATA" 테이블  
테이블명을 클릭해 내부에 저장된 테이블 데이터들을 확인할 수 있다.**

© AFI, Inc. All rights reserved.



# 게임 정보 추가

## ■ 유저 정보 테이블 생성 (계속)

Backnd Console



ProjectA



관리자



ProjectA 개발 모드

★ 게임 정보 관리 + 행 생성 ↶ 삭제 ↶ 테이블 초기화

[SDK 문서](#)

[콘솔 가이드](#)

★ 즐겨찾기

대시보드

서버 설정

프로젝트

인증 정보

푸시

스토어 정보

소셜

예산 알림

요금제

뒤끝베이스

유저 관리

유저 접근 관리

게임 정보 관리

랭킹 관리

우편 관리

푸시 관리

쿠폰 관리

차트 관리

테이블    데이터

테이블 선택

USER\_DATA

상세 검색 ▶

검색

검색 조건 초기화

데이터가 없습니다.

**방금 생성했기 때문에 데이터가 비어 있다.**



# 게임 정보 추가

## ■ 유니티에 유저 정보 클래스 생성

- 유저의 게임 내 정보 변수들을 선언하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "UserData"로 변경

```
1  [System.Serializable]
2  public class UserData
3  {
4      public int    level;           // Lobby Scene에 보이는 플레이어 레벨
5      public float  experience;     // Lobby Scene에 보이는 플레이어 경험치
6      public int    gold;           // 무료 재화
7      public int    jewel;         // 유료 재화
8      public int    heart;         // 게임 플레이에 소모되는 재화
9
10     public void Reset()
11     {
12         level      = 1;
13         experience = 0;
14         gold       = 0;
15         jewel      = 0;
16         heart      = 30;
17     }
18 }
```





# 게임 정보 추가

## ■ 계정을 생성할 때 테이블에 유저 정보 추가

Insert() : 테이블에 row를 추가

해당 row는 insert 한 유저에게 귀속된다.

스키마 정의/미정의 여부 관계없이 데이터를 삽입할 수 있다.

public/private 여부 관계없이 데이터를 삽입할 수 있다.

스키마 삽입 시, 값이 비었을 경우 null 또는 기본값으로 입력된다.

숫자형 데이터의 경우, 최대 9007199254740991(약 9000조)까지 안전하게 값을 넣을 수 있습니다.

그 이상의 데이터를 삽입할 경우, 일의 자리, 십의 자리 수는 0으로 내림처리가 됩니다.

숫자가 증가하여 자리 수가 많아질수록 작은 수는 내림처리가 되며 큰 수는 왼쪽 기준으로 16~18자리의 수까지 값이 유지됩니다.



# 게임 정보 추가

- 서버와 연동해 유저 정보를 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "BackendGameData"로 변경

```
1  using UnityEngine;
2  using BackEnd;
3
4  public class BackendGameData
5  {
6      private static BackendGameData instance = null;
7      public static BackendGameData Instance
8      {
9          get
10         {
11             if ( instance == null )
12             {
13                 instance = new BackendGameData();
14             }
15
16             return instance;
17         }
18     }
19
```



# 게임 정보 추가

- 서버와 연동해 유저 정보를 제어하는 스크립트 생성 및 작성 (계속)

```
20     private UserData userGameData = new UserData();
21     public UserData UserData => userGameData;
22
23     private string gameDataRowInDate = string.Empty;
24
25     /// <summary>
26     /// 뒤끝 콘솔 테이블에 새로운 유저 정보 추가
27     /// </summary>
28     public void GameDataInsert()
29     {
30         // 유저 정보를 초기값으로 설정
31         userGameData.Reset();
32
33         // 테이블에 추가할 데이터로 가공
34         Param param = new Param()
35         {
36             { "level",      userGameData.level },
37             { "experience", userGameData.experience },
38             { "gold",      userGameData.gold },
39             { "jewel",     userGameData.jewel },
40             { "heart",     userGameData.heart }
41         };
42
```



# 게임 정보 추가

- 서버와 연동해 유저 정보를 제어하는 스크립트 생성 및 작성 (계속)

```
43 // 첫 번째 매개변수는 뒤끝 콘솔의 "게임 정보 관리" 탭에 생성한 테이블 이름
44 Backend.GameData.Insert("USER_DATA", param, callback =>
45 {
46     // 게임 정보 추가에 성공했을 때
47     if ( callback.IsSuccess() )
48     {
49         // 게임 정보의 고유값
50         gameDataRowInDate = callback.GetInDate();
51
52         Debug.Log($"게임 정보 데이터 삽입에 성공했습니다. : {callback}");
53     }
54     // 실패했을 때
55     else
56     {
57         Debug.LogError($"게임 정보 데이터 삽입에 실패했습니다. : {callback}");
58     }
59 });
60 }
61 }
```



# 게임 정보 추가

- 계정 생성에 성공했을 때 서버 테이블에 해당 계정 유저 정보 생성
  - RegisterAccount Script 수정

```
64  // <summary> 계정 생성 시도 후 서버로부터 전달받은 message를 기반으로 로직 처리
67  private void CustomSignUp()
68  {
69      Backend.BMember.CustomSignUp(inputFieldID.text, inputFieldPW.text, callback =>
70      {
71          // "계정 생성" 버튼 상호작용 활성화
72          btnRegisterAccount.interactable = true;
73
74          // 계정 생성 성공
75          if ( callback.IsSuccess() )
76          {
77              // E-mail 정보 업데이트
78              Backend.BMember.UpdateCustomEmail(inputFieldEmail.text, callback =>
79              {
80                  if ( callback.IsSuccess() )
81                  {
82                      SetMessage($"계정 생성 성공. {inputFieldID.text}님 환영합니다.");
83
84                      // 계정 생성에 성공했을 때 해당 계정의 게임 정보 생성
85                      BackendGameData.Instance.GameDataInsert();
86
87                      // Lobby 씬으로 이동
88                      Utils.LoadScene(SceneNames.Lobby);
89                  }
90              });
91          }
92          // 계정 생성 실패
93          else...
121  });
122  }
123  }
```



# 게임 정보 추가

## ■ 결과 화면

The screenshot shows a Unity 2022.2.21f1 console window displaying a web application. The browser address bar shows the URL: `console.thebackend.io/project/1ea3f14d34e89530ea88b3245bc82dc17d5f52ce1554049f19fce9219a847cfce18bb8891bc9af94bc66efb9a3b7878c6bc5513c1dd300af990c59a6ed...`. The web interface is titled "ProjectA" and features a sidebar menu with various game management options. The main content area is titled "게임 정보 관리" (Game Information Management) and includes tabs for "게임 정보 관리", "통계성", "삭제", and "테이블 초기화". A dropdown menu is set to "USER\_DATA". Below the dropdown, there are buttons for "검색" (Search) and "검색 조건 초기화" (Reset Search Conditions). The message "데이터가 없습니다." (No data) is displayed. The footer of the web interface includes "회사소개", "이용약관", "서비스수준협약", "개인정보처리방침", and "© AFI, Inc. All rights reserved."

# 게임 정보 불러오기

- Lobby Scene UI 제작
- Lobby 씬을 로드할 때 유저 정보 불러오기





# 게임 정보 불러오기

## ■ Lobby Scene UI 제작

### ■ 플레이어 레벨을 출력하는 "Text - TextMeshPro" UI 생성 및 설정

#### □ GameObject - UI - "Text - TextMeshPro"

The screenshot shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Level' object is selected under the 'UserInfo' folder. The Inspector panel shows the 'Rect Transform' component with the following values:

Property	Value
Pos X	-230
Pos Y	-40
Pos Z	0
Width	70
Height	40

The 'Anchors' section is also visible, showing the following values:

Property	X	Y
Min	1	0.5
Max	1	0.5
Pivot	1	0.5

The screenshot shows the TextMeshPro - Text (UI) component inspector. The 'Text Input' field contains the text '999'. The 'Main Settings' section is highlighted with a red box, showing the following values:

Property	Value
Font Asset	NotoSansKR-Regular SDF (TMF)
Material Preset	NotoSansKR-Regular SDF Material
Font Style	B
Font Size	30

The 'Alignment' section is also highlighted with a red box, showing the 'Center' alignment icon selected.





# 게임 정보 불러오기

- 플레이어 경험치를 출력하는 Slider UI 생성 및 설정
  - GameObject - UI - Slider

The screenshot displays the Unity Inspector for a Slider UI element. The Hierarchy panel on the left shows the object structure: Lobby\* > Canvas > TopPanel > UserInfo > Experience. The Inspector panel on the right shows the following settings:

- Rect Transform**
  - Position: Pos X: -20, Pos Y: -40, Pos Z: 0
  - Width: 210, Height: 40
  - Min Anchor: X 1, Y 0.5
  - Max Anchor: 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
- Slider**
  - Interactable:
  - Transition: Color Tint
  - Target Graphic: Handle (Image)
  - Normal Color: [Color Picker]
  - Highlighted Color: [Color Picker]



# 게임 정보 불러오기

- 플레이어 경험치를 출력하는 Slider UI 생성 및 설정 (계속)

The screenshot displays the Unity Inspector window with the following settings:

- Background** (Static): Tag: Untagged, Layer: UI
- Rect Transform**:
  - stretch: stretch (checked)
  - Left: 0, Top: 0, Pos Z: 0
  - Right: 0, Bottom: 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**:
  - Source Image: Background
  - Color: (Color picker)
  - Material: None (Material) **Color (0, 0, 0, 255)**
  - Raycast Target: checked
  - Raycast Padding: (disabled)



# 게임 정보 불러오기

- 플레이어 경험치를 출력하는 Slider UI 생성 및 설정 (계속)

The screenshot shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby\*' with components like 'Main Camera', 'Canvas', 'TopPanel', 'UserInfo', 'ProfileFrame', 'NicknameBackground', 'Level', 'Experience', 'Background', 'Fill Area', and 'Fill'. The 'Fill Area' component is selected and highlighted with a red box. The Inspector panel on the right shows the properties of the 'Fill Area' component, which is a 'Rect Transform' component. The 'Rect Transform' component is also highlighted with a red box. The 'Rect Transform' component has the following properties:

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

The 'Fill Area' component is also highlighted with a red box. The 'Fill Area' component has the following properties:

Property	Value
Fill Area	Static
Tag	Untagged
Layer	UI

The 'Fill Area' component is also highlighted with a red box. The 'Fill Area' component has the following properties:

Property	Value
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



# 게임 정보 불러오기

- 플레이어 경험치를 출력하는 Slider UI 생성 및 설정 (계속)

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure for a 'Lobby' scene. The 'Experience' object is selected, and its 'Fill' component is highlighted with a red box. The Inspector panel on the right shows the properties for the 'Fill' component, which is a 'Rect Transform'.

**Rect Transform Properties:**

- Some values driven by Slider.
- left:
- Pos X: 0, Top: 0, Pos Z: 0
- Width: 0, Bottom: 0
- Anchors: Min (X: 0, Y: 0), Max (X: 0, 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 Properties:**

- Image: Source Image (UISprite), Color (Red), Material (None (Material Color (0, 255, 0, 255))), Raycast Target (checked)



# 게임 정보 불러오기

- 플레이어의 재화 UI를 관리하는 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 'Canvas' child contains 'TopPanel', which in turn contains 'UserInfo' and 'Goods'. The 'Goods' object is selected and highlighted with a red box. On the right, the Inspector panel shows the 'Goods' object selected. The 'Rect Transform' component is expanded and highlighted with a red box, showing its position (Pos X: 0, Pos Y: -25, Pos Z: 0), width (960), and height (100). Below it, the 'Canvas Renderer' component is visible, and the 'Image' component is expanded, with its 'Source Image' field set to 'Background' and highlighted with a red box.



# 게임 정보 불러오기

- 게임 플레이 시 소모되는 하트 UI를 관리하는 Button UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

Button 컴포넌트에 Collider가 있어 충돌이 가능하고, 화면에 출력은 자식 오브젝트들이 하기 때문에 CanvasRenderer, Image Component를 삭제해서 보이지 않도록 설정



# 게임 정보 불러오기

- 게임 플레이 시 소모되는 하트 UI를 관리하는 Button UI 생성 및 설정 (계속)

버튼과 함께 생성된 Text (TMP)  
오브젝트의 이름을 "Time" 으로 변경

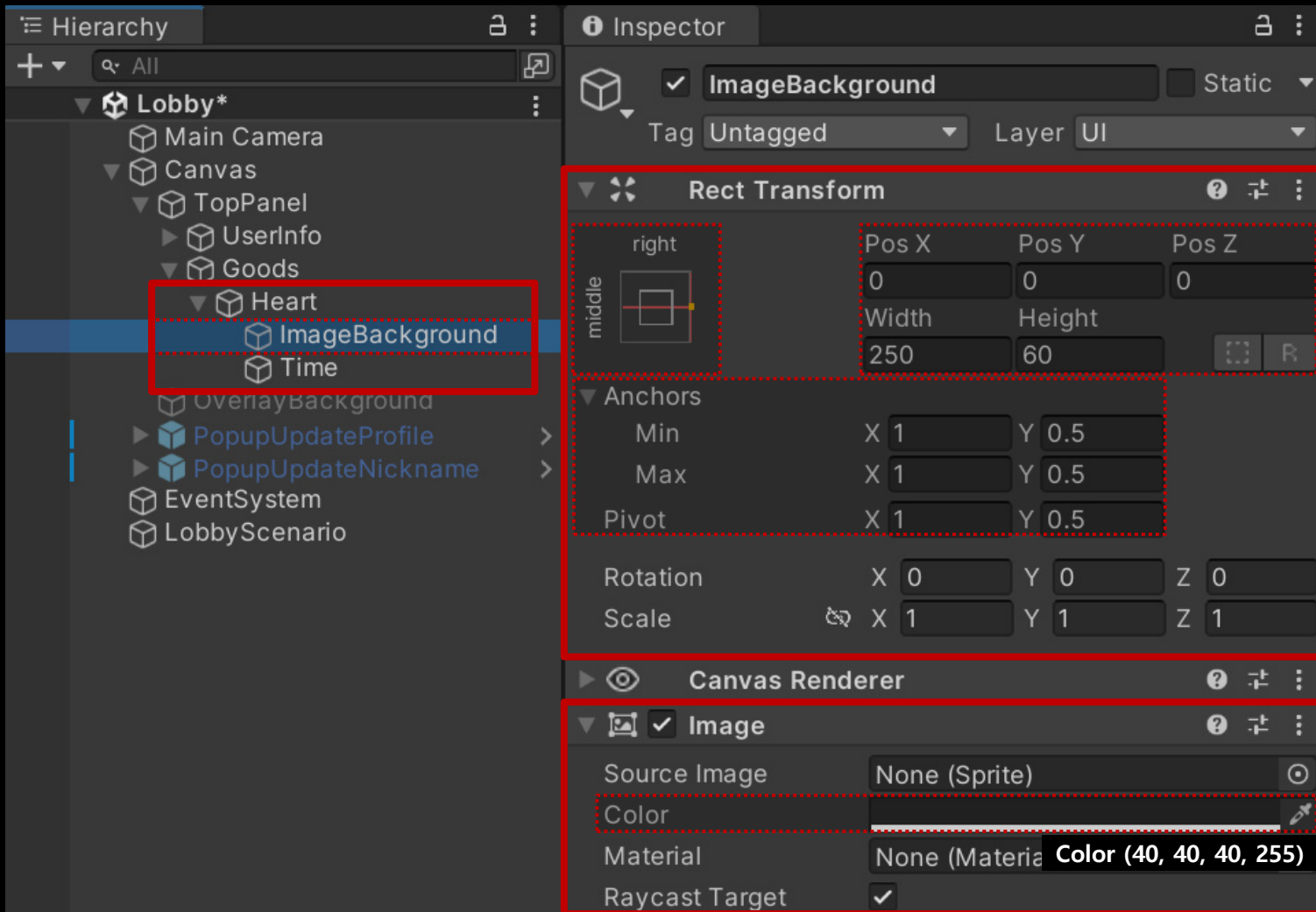
Color (255, 255, 255, 255)





# 게임 정보 불러오기

- 하트 개수 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image







# 게임 정보 불러오기

- 하트 개수를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Text - TextMeshPro"

The screenshot shows the Unity interface. In the Hierarchy panel, the 'HeartCount' object is selected under the 'ImageBackground' parent. The Inspector panel displays the 'Rect Transform' component with a stretch box and 'Anchors' set to Min (0,0) and Max (1,1). The 'TextMeshPro - Text (UI)' component is also visible below.

The screenshot shows the TextMeshPro Inspector panel. The 'Text Input' field contains '10/30'. The 'Main Settings' section shows 'Font Asset' as 'NotoSansKR-Bold SDF (TMP\_Fc...', 'Font Style' as 'B', 'Font Size' as '36', and 'Alignment' as 'Center'.



# 게임 정보 불러오기

- 하트 이미지를 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

The screenshot displays the Unity Inspector for a GameObject named 'ImageHeart'. The 'Rect Transform' component is highlighted with a red dashed box, showing its position (0, 0, 0) and size (100, 100). The 'Image' component is also highlighted, with its 'Source Image' set to 'UI\_TopPanel\_Icon\_Heart'. The Hierarchy panel on the left shows the object's placement within the 'Lobby' scene, specifically under 'Canvas' > 'TopPanel' > 'Heart' > 'ImageHeart'. The Project panel on the right shows the 'Assets' folder structure, with 'UI\_TopPanel\_Icon\_Heart' highlighted in the 'Assets > Textures > UI' path.

하트 이미지가 하트 개수 배경 (ImageBackground)보다 앞에 출력되어야 하기 때문에 더 밑에 배치



# 게임 정보 불러오기

- 유료 재화 보석 UI를 관리하는 Button UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Lobby\*' folder is expanded, showing a tree structure: Main Camera, Canvas, TopPanel, UserInfo, Goods, Heart, Jewel, and Text (TMP). The 'Jewel' object is selected and highlighted in blue. The Inspector panel shows the properties for the selected 'Jewel' object. The 'Rect Transform' component is visible, with the pivot set to 'center' and 'middle'. The position (Pos X, Pos Y, Pos Z) is set to (0, 0, 0). The width is 300 and the height is 100. The 'Image' component is also visible, with the source image set to 'UISprite', the color set to white, and the material set to 'None (Material)'. A red dashed box highlights the 'Rect Transform' and 'Image' components.



# 게임 정보 불러오기

- 보석 개수 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

The screenshot displays the Unity Inspector for a GameObject named 'ImageBackground'. The Hierarchy panel on the left shows the object's path: Lobby\* > Canvas > TopPanel > Goods > Jewel > ImageBackground. The Inspector panel shows the following settings:

- ImageBackground** (Tag: Untagged, Layer: UI)
- Rect Transform**
  - Position: Pos X: 0, Pos Y: 0, Pos Z: 0
  - Width: 250, Height: 60
  - 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** (Source Image: None (Sprite), Color: Color (40, 40, 40, 255), Raycast Target: checked)



# 게임 정보 불러오기

- 보석 개수를 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 'Lobby\*' object with a 'JewelCount' child. The Inspector panel on the right shows the 'JewelCount' component with a 'Rect Transform' component highlighted in red. The 'Rect Transform' component has a 'stretch' property set to 'stretch' and 'Left' set to 30. The 'Anchors' section is also highlighted in red, showing 'Min' at (0,0) and 'Max' at (1,1).

The screenshot shows the TextMeshPro Inspector panel. The 'Text Input' field contains '2222'. The 'Main Settings' section is highlighted in red, showing 'Font Asset' as 'NotoSansKR-Bold SDF (TMP\_Fc...', 'Material Preset' as 'NotoSansKR-Bold SDF Material', and 'Font Style' as 'B'. The 'Alignment' section is also highlighted in red, showing the 'Left' alignment icon selected.



# 게임 정보 불러오기

- 보석 이미지를 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

보석 이미지가 보석 개수 배경 (ImageBackground)보다 앞에 출력되어야 하기 때문에 더 밑에 배치

Component	Property	Value
Rect Transform	Pos X	0
	Pos Y	5
	Pos Z	0
	Width	100
	Height	100
	Min X	0
	Min Y	0.5
	Max X	0
	Max Y	0.5
	Pivot X	0
Pivot Y	0.5	
Rotation X	0	
Rotation Y	0	
Rotation Z	0	
Scale X	1	
Scale Y	1	
Scale Z	1	
Canvas Renderer	Order in Layer	0
	Material	None (Material)
Image	Source Image	UI_TopPanel_Icon_Jewel
	Color	[Color Picker]
	Material	None (Material)
	Raycast Target	✓
	Raycast Padding	[Value]
	Image Type	Simple





# 게임 정보 불러오기

- 무료 재화 골드 UI를 관리하는 Button UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

The screenshot displays the Unity development environment. On the left, the Hierarchy panel shows a tree structure under 'Lobby\*'. The 'Gold' object is selected, and its 'Text (TMP)' component is highlighted with a red dashed box. On the right, the Inspector panel shows the properties of the selected object. The 'Rect Transform' component is expanded, showing 'Pos X' set to -20, 'Pos Y' to 0, 'Width' to 300, and 'Height' to 100. The 'Image' component is also expanded, showing 'Source Image' set to 'UISprite'. The 'Image' component's 'Color' property is set to white, and 'Raycast Target' is checked.



# 게임 정보 불러오기

- 골드 개수 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

The screenshot displays the Unity Inspector window for a GameObject named 'ImageBackground' under the 'Lobby\*' hierarchy. The 'Rect Transform' component is selected, showing its position (0, 0, 0), width (250), and height (60). The 'Anchors' are set to Min (X: 1, Y: 0.5) and Max (X: 1, Y: 0.5), with a Pivot at (X: 1, Y: 0.5). The 'Canvas Renderer' component is also visible. The 'Image' component is selected, showing the 'Color' property set to (40, 40, 40, 255).

Component	Property	Value
Rect Transform	Pos X	0
	Pos Y	0
	Pos Z	0
	Width	250
Rect Transform	Height	60
	Anchor Min X	1
Anchor Min	Anchor Min Y	0.5
	Anchor Max X	1
Anchor Max	Anchor Max Y	0.5
	Pivot X	1
Pivot	Pivot Y	0.5
	Rotation X	0
Rotation	Rotation Y	0
	Rotation Z	0
	Scale X	1
Scale	Scale Y	1
	Scale Z	1
	Canvas Renderer	Source Image
Color		(40, 40, 40, 255)
Image	Material	None (Material)
	Raycast Target	✓





# 게임 정보 불러오기

- 골드 개수를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Text - TextMeshPro"

The screenshot shows the Unity interface. In the Hierarchy panel, the 'GoldCount' object is selected under the 'ImageBackground' parent. The Inspector panel displays the 'Rect Transform' component with a stretch box and 'Anchors' set to Min (0,0) and Max (1,1). The 'TextMeshPro - Text (UI)' component is also visible in the Inspector panel.

The screenshot shows the TextMeshPro Inspector panel. The 'Text Input' field contains '222K'. The 'Main Settings' section shows 'Font Asset' as 'NotoSansKR-Bold SDF (TMP\_Fc...)', 'Font Style' as 'B I U S ab AB SC', and 'Font Size' as '36'. The 'Alignment' section shows 'Left' alignment selected.



# 게임 정보 불러오기

- 골드 이미지를 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

Inspector: ImageGold (Static, Tag: Untagged, Layer: UI)

Rect Transform: left, middle, Pos X: 0, Pos Y: 0, Pos Z: 0, Width: 100, Height: 100

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

Canvas Renderer

Image: Source Image: UI\_TopPanel\_Icon\_Gold, Color: [Color Picker], Material: None (Material), Raycast Target: [checked], Maskable: [checked], Image Type: Simple

Hierarchy: Lobby\* > Canvas > TopPanel > Gold > ImageBackground > ImageGold

Assets > Textures > UI: UI\_TopPanel\_Icon\_Gold

골드 이미지가 골드 개수 배경 (ImageBackground)보다 앞에 출력되어야 하기 때문에 더 밑에 배치



# 게임 정보 불러오기

## ■ Lobby 씬을 로드할 때 유저 정보 불러오기

**GetMyData() : 자기 자신의 데이터를 조회**

스키마 정의/미정의 여부 관계없이 데이터를 조회할 수 있다.

public/private 여부 관계없이 자기 자신의 데이터만 조회할 수 있다.

select 절의 사용 유무와 관계없이 데이터의 처리량은 동일하다.

**new Where() : 조건 없이 모든 데이터 조회**

조건 없이 모든 데이터를 검색하고자 할 때는 인자 값으로 넘기는 where을 new Where()로 선언해 주면 모든 데이터를 검색할 수 있다.



# 게임 정보 불러오기

- 뒤끝 콘솔 테이블에 저장되어 있는 유저 정보 불러오기
  - BackendGameData Script 수정

```
1  using UnityEngine;
2  using BackEnd;
3
4  public class BackendGameData
5  {
6      [System.Serializable]
7      public class GameDataLoadEvent : UnityEngine.Events.UnityEvent { }
8      public GameDataLoadEvent onGameDataLoadEvent = new GameDataLoadEvent();
9
10     private static BackendGameData instance = null;
11     public static BackendGameData Instance...
12
13
14     private UserData userData = new UserData();
15     public UserData UserData => userData;
16
17     private string gameDataRowInDate = string.Empty;
18
19     /// <summary> 뒤끝 콘솔 테이블에 새로운 유저 정보 추가
20     public void GameDataInsert()...
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
```



# 게임 정보 불러오기

## BackendGameData Script 수정 (계속)

```
66  /// <summary>
67  /// 뒤끝 콘솔 테이블에서 유저 정보를 불러올 때 호출
68  /// </summary>
69  public void GameDataLoad()
70  {
71      Backend.GameData.GetMyData("USER_DATA", new Where(), callback =>
72      {
73          // 게임 정보 불러오기에 성공했을 때
74          if ( callback.IsSuccess() )
75          {
76              Debug.Log($"게임 정보 데이터 불러오기에 성공했습니다. : {callback}");
77
78              // JSON 데이터 파싱 성공
79              try{...}
110         }
111         // 실패했을 때
112         else
113         {
114             Debug.LogError($"게임 정보 데이터 불러오기에 실패했습니다. : {callback}");
115         }
116     });
117 }
118 }
```

뒷장





# 게임 정보 불러오기

## BackendGameData Script 수정 (계속)

```
78 // JSON 데이터 파싱 성공
79 try
80 {
81     LitJson.JsonData gameDataJson = callback.FlattenRows();
82
83     // 받은 데이터의 개수가 0이면 데이터가 없는 것
84     if ( gameDataJson.Count <= 0 )
85     {
86         Debug.LogWarning("데이터가 존재하지 않습니다.");
87     }
88     else
89     {
90         // 불러온 게임 정보의 고유값
91         gameDataRowInDate = gameDataJson[0]["inDate"].ToString();
92         // 불러온 게임 정보를 userData 변수에 저장
93         userGameData.level = int.Parse(gameDataJson[0]["level"].ToString());
94         userGameData.experience = float.Parse(gameDataJson[0]["experience"].ToString());
95         userGameData.gold = int.Parse(gameDataJson[0]["gold"].ToString());
96         userGameData.jewel = int.Parse(gameDataJson[0]["jewel"].ToString());
97         userGameData.heart = int.Parse(gameDataJson[0]["heart"].ToString());
98
99         onGameDataLoadEvent?.Invoke();
100     }
101 }
102 // JSON 데이터 파싱 실패
103 catch ( System.Exception e )
104 {
105     // 유저 정보를 초기값으로 설정
106     userGameData.Reset();
107     // try-catch 에러 출력
108     Debug.LogError(e);
109 }
```



# 게임 정보 불러오기

- 서버에서 불러온 유저 정보 출력 (레벨, 경험치 등)
  - TopPanelViewer Script 수정

```
1  using UnityEngine;
2  using UnityEngine.UI;
3  using TMPro;
4
5  public class TopPanelViewer : MonoBehaviour
6  {
7      [SerializeField]
8      private TextMeshProUGUI textNickname;
9      [SerializeField]
10     private TextMeshProUGUI textLevel;
11     [SerializeField]
12     private Slider          sliderExperience;
13     [SerializeField]
14     private TextMeshProUGUI textHeart;
15     [SerializeField]
16     private TextMeshProUGUI textJewel;
17     [SerializeField]
18     private TextMeshProUGUI textGold;
19 }
```



# 게임 정보 불러오기

- TopPanelViewer Script 수정 (계속)

```
20 private void Awake()  
21 {  
22     BackendGameData.Instance.onGameDataLoadEvent.AddListener(UpdateGameData);  
23 }  
24  
25 public void UpdateNickname()...  
31  
32 public void UpdateGameData()  
33 {  
34     textLevel.text = $"{BackendGameData.Instance.UserGameData.level}";  
35     // 임시로 최대 경험치를 100으로 설정  
36     sliderExperience.value = BackendGameData.Instance.UserGameData.experience / 100;  
37     textHeart.text = $"{BackendGameData.Instance.UserGameData.heart} / 30";  
38     textJewel.text = $"{BackendGameData.Instance.UserGameData.jewel}";  
39     textGold.text = $"{BackendGameData.Instance.UserGameData.gold}";  
40 }  
41 }
```





# 게임 정보 불러오기

- TopPanel 오브젝트의 "TopPanelViewer" 컴포넌트 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the 'Lobby\*' object selected, with 'TopPanel' highlighted. The Inspector panel on the right shows the 'TopPanelViewer (Script)' component. Red boxes and arrows indicate the mapping of variables in the script to corresponding objects in the hierarchy.

Inspector Variable	Hierarchy Object
Text Nickname	NicknameBackground
Text Level	Level
Slider Experience	Experience
Text Heart	HeartCount
Text Jewel	JewelCount
Text Gold	GoldCount



# 게임 정보 불러오기

- Lobby 씬을 로드할 때 서버로부터 유저 정보 불러오기
  - LobbyScenario Script 수정

```
1  using UnityEngine;
2
3  public class LobbyScenario : MonoBehaviour
4  {
5      [SerializeField]
6      private UserInfo user;
7
8      private void Awake() ...
12
13  private void Start()
14  {
15      BackendGameData.Instance.GameDataLoad();
16  }
17 }
```



# 게임 정보 불러오기

## ■ 결과 화면

UI Screenshot showing a game information table and a login dialog box.

UI Elements: # Scene, Game, Animator, Animation, Game, Display 1, 1440x2960, Scale 0.39x, Play Focused, Stats, Gizmos

gamer_id	inDate	owner_inDate	updatedAt	client_date	exp...	gold	heart	jewel	level
<a href="#">0c0fa510-ab62-11ed-a7ef-859caee01a10</a>	2023-02-13T05:48:32.439Z	2023-02-13T05:48:31.585Z	2023-02-13T05:48:32.439Z	2023-02-13T05:48:31.138Z	0	0	30	0	1

번호	회원번호	회원 아이디	닉네임	가입일	최종 접속일	국가	접속 OS
1	<a href="#">0c0fa510-ab62-11ed-a7ef-859caee01a10</a>	user04	-	2023.02.13 14:48	2023.02.13 15:58	-	Windows 10 &#4010...

**로그인**

아이디

비밀번호

**로그인**

[아이디 찾기](#)   [계정 생성](#)   [비밀번호 찾기](#)