

# Unity Backend

“랭킹” 기능을 이용한 유저 랭킹 관리 (랭킹 출력)

Created in 2023-06-07  
Last Updated 2023-06-09  
Unity Version 2022.2.2f1

## *Index*

- ◆ 랭킹 데이터 불러오기

# 랭킹 데이터 불러오기

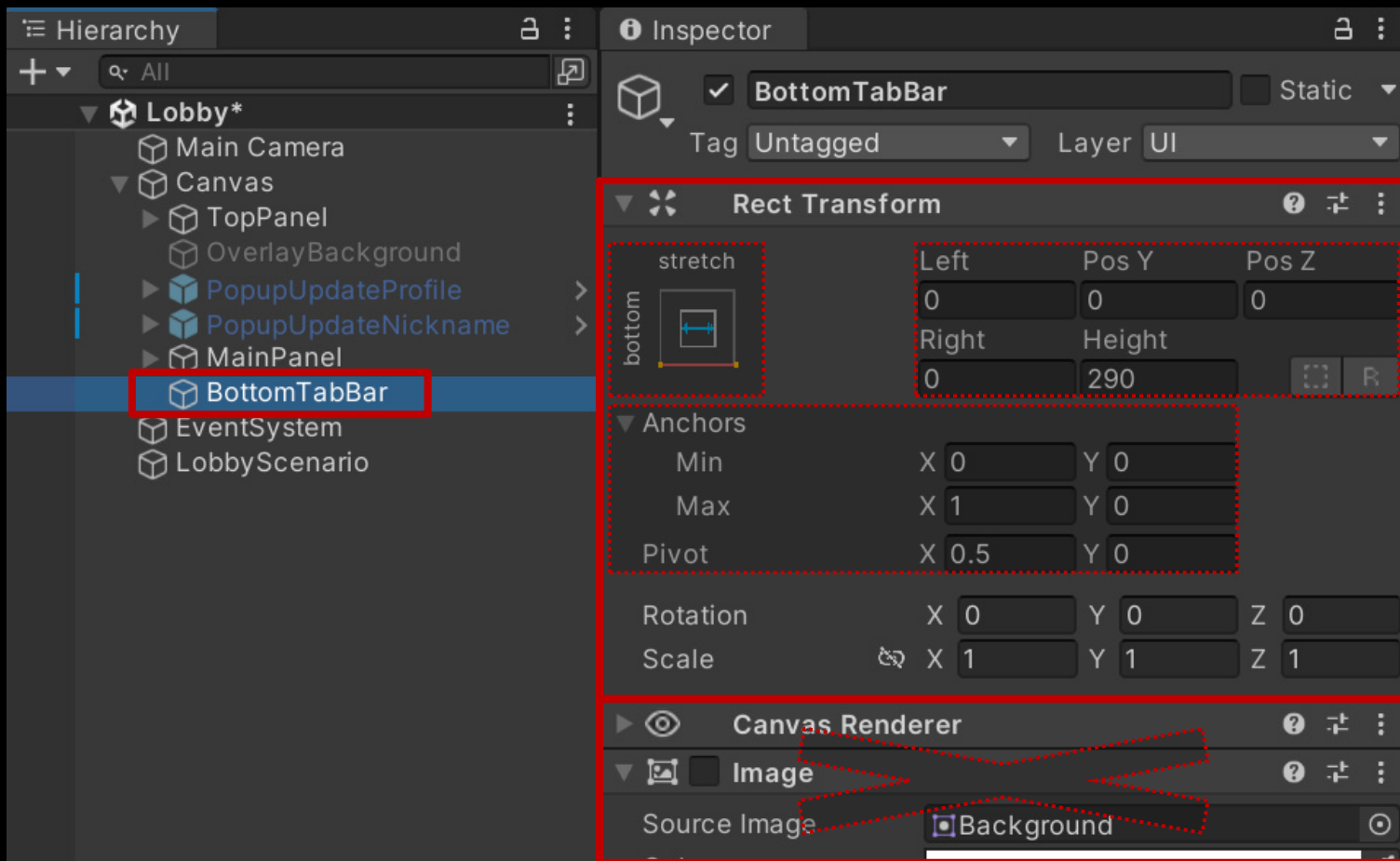
- Lobby 씬 TabBar 제작
- Rank 탭 제작
- 랭킹 데이터 불러오기



# 랭킹 데이터 불러오기

## ■ Lobby 씬 TabBar 제작

- TabBar UI를 관리하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel





# 랭킹 데이터 불러오기

- BottomTabBar 오브젝트에 컴포넌트 추가 및 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **BottomTabBar** component is selected under the **Lobby\*** object. The Inspector panel shows the following configuration for the **BottomTabBar** component:

- BottomTabBar** (Static: )
- Tag: Untagged, Layer: UI
- Rect Transform**
- Horizontal Layout Group** (Selected):
  - Padding: Left: 5, Right: 5, Top: 5, Bottom: 5
  - Spacing: 5
  - Child Alignment: Middle Center
  - Reverse Arrangement:
  - Control Child Size:  Width,  Height
  - Use Child Scale:  Width,  Height
  - Child Force Expand:  Width,  Height
- Toggle Group** (Selected):
  - Script:  ToggleGroup
  - Allow Switch Off:

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



# 랭킹 데이터 불러오기

- Shop Toggle UI 생성 및 설정
  - GameObject - UI - Toggle

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

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - MainPanel
    - BottomTabBar
      - Shop (Selected)
      - Background
        - Checkmark
        - Label (Crossed out)
    - EventSystem
    - LobbyScenario

**Inspector Panel:**

- Shop
  - Tag: Untagged
  - Layer: UI
  - Rect Transform
    - Some values driven by HorizontalLayoutGroup.
    - left: Pos X 720, Pos Y -145, Pos Z 0
    - top: Width 1430, Height 280
    - Anchors
      - Min: X 0, Y 1
      - 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
  - Toggle

**Text Box:** Label은 사용하지 않기 때문에 삭제



# 랭킹 데이터 불러오기

## ■ Shop Toggle UI 생성 및 설정 (계속)

The screenshot shows the Unity Inspector for a UI element named 'Background'. The 'Rect Transform' component is highlighted with a red dashed box, showing 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

The 'Image' component is also highlighted with a red dashed box, showing the following settings:

Source Image	UI_TabBar_Shop
Color	Color (100, 100, 100, 255)
Material	No Material
Raycast Target	Checked
Raycast Padding	0
Maskable	Checked
Image Type	Simple

The 'Project' panel on the right shows the 'Assets > Textures > UI' folder structure, with 'UI\_TabBar\_Shop' highlighted in red. A red arrow points from this asset to the 'Source Image' field in the Inspector.



# 랭킹 데이터 불러오기

## ■ Shop Toggle UI 생성 및 설정 (계속)

The screenshot shows the Unity Inspector for a 'Checkmark' UI element. The 'Rect Transform' component is highlighted with a red dashed box, showing 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

The 'Image' component is also highlighted with a red dashed box, showing the following settings:

Source Image	UI_TabBar_Shop
Color	[Color Field]
Material	None (Material)
Raycast Target	✓
Raycast Padding	[Field]
Maskable	✓
Image Type	Simple

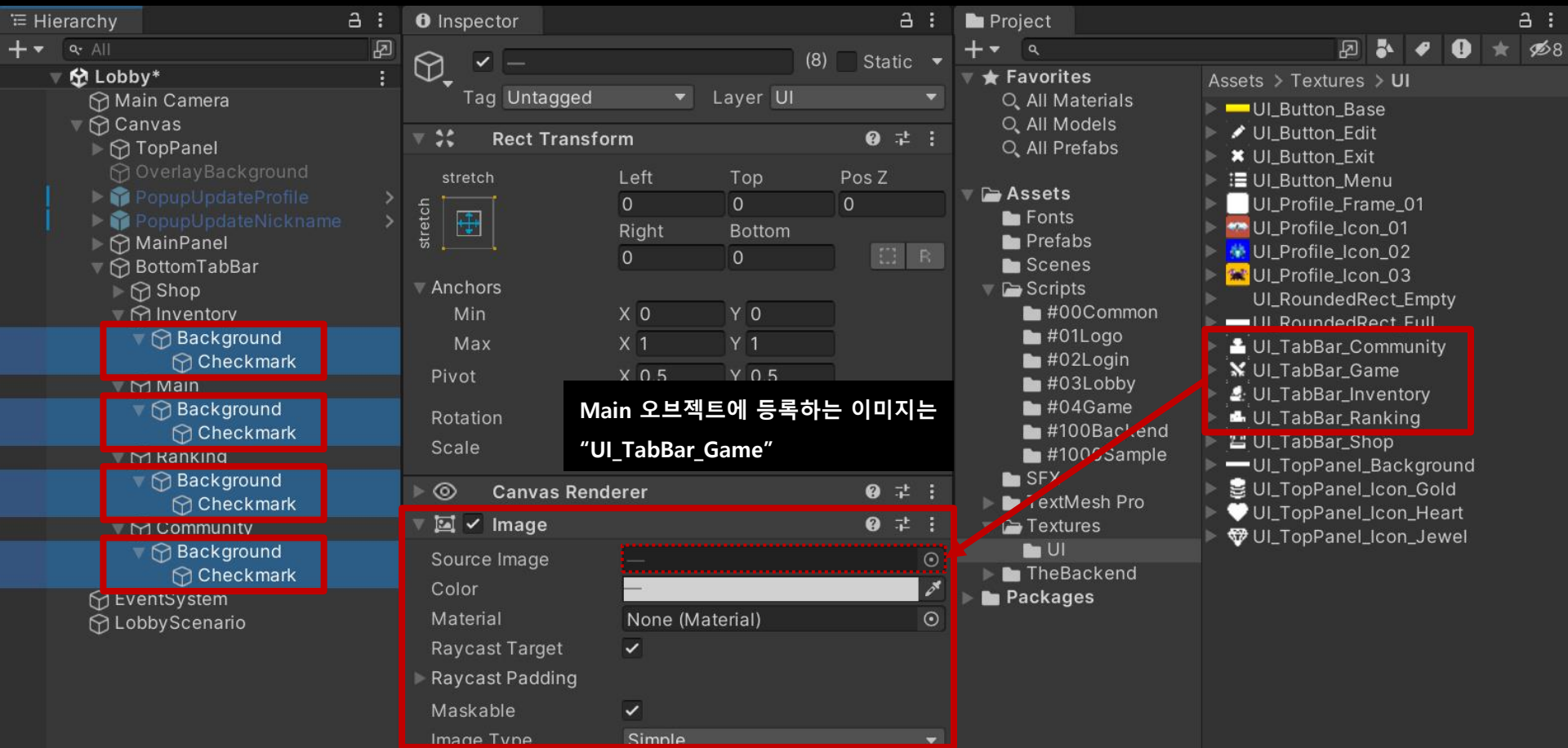
The 'Assets' panel on the right shows the 'UI\_TabBar\_Shop' texture selected in the 'Image Source' field. The 'Hierarchy' panel on the left shows the 'Checkmark' object selected under the 'Shop' folder.





# 랭킹 데이터 불러오기

- Inventory, Main, Ranking, Community Toggle UI 생성 및 설정
  - Shop 오브젝트를 Ctrl+D로 복제





# 랭킹 데이터 불러오기

- Toggle 오브젝트들의 "Toggle" 컴포넌트 Group 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Lobby\*' object is expanded, showing a 'BottomTabBar' child. The Inspector panel shows the 'Toggle' component for 'BottomTabBar' with various settings. A red box highlights the 'BottomTabBar' in the Hierarchy and the 'Toggle' component in the Inspector. A red arrow points from the 'BottomTabBar' in the Hierarchy to the 'BottomTabBar (Toggle Group)' in the Inspector's Group list.

Property	Value
Interactable	<input checked="" type="checkbox"/>
Transition	Color Tint
Target Graphic	—
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	<input checked="" type="checkbox"/>
Toggle Transition	Fade
Graphic	—
Group	<input checked="" type="checkbox"/> BottomTabBar (Toggle Group)



# 랭킹 데이터 불러오기

- Main을 제외한 Toggle 오브젝트들의 IsOn 변수 설정

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left displays a tree structure under 'Lobby\*'. The Inspector panel on the right shows the 'Toggle' component settings for a selected object. The 'Is On' property is highlighted with a red dashed box.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - MainPanel
    - BottomTabBar
      - Shop
      - Inventory
      - Main
      - Ranking
      - Community
  - EventSystem
  - LobbyScenario

**Inspector Panel:**

- Tag: Untagged
- Layer: UI
- Rect Transform
- Toggle**
  - Interactable:
  - Transition: Color Tint
    - Target Graphic: —
    - 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: —
  - Group:  BottomTabBar (Toggle Group)



# 랭킹 데이터 불러오기

- 각 탭의 페이지를 관리하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel

1. 생성한 Panel의 이름을 TabBarPages로 변경  
2. MainPanel 오브젝트의 이름을 MainPage로 변경하고, TabBarPages 오브젝트의 자식으로 설정



# 랭킹 데이터 불러오기

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

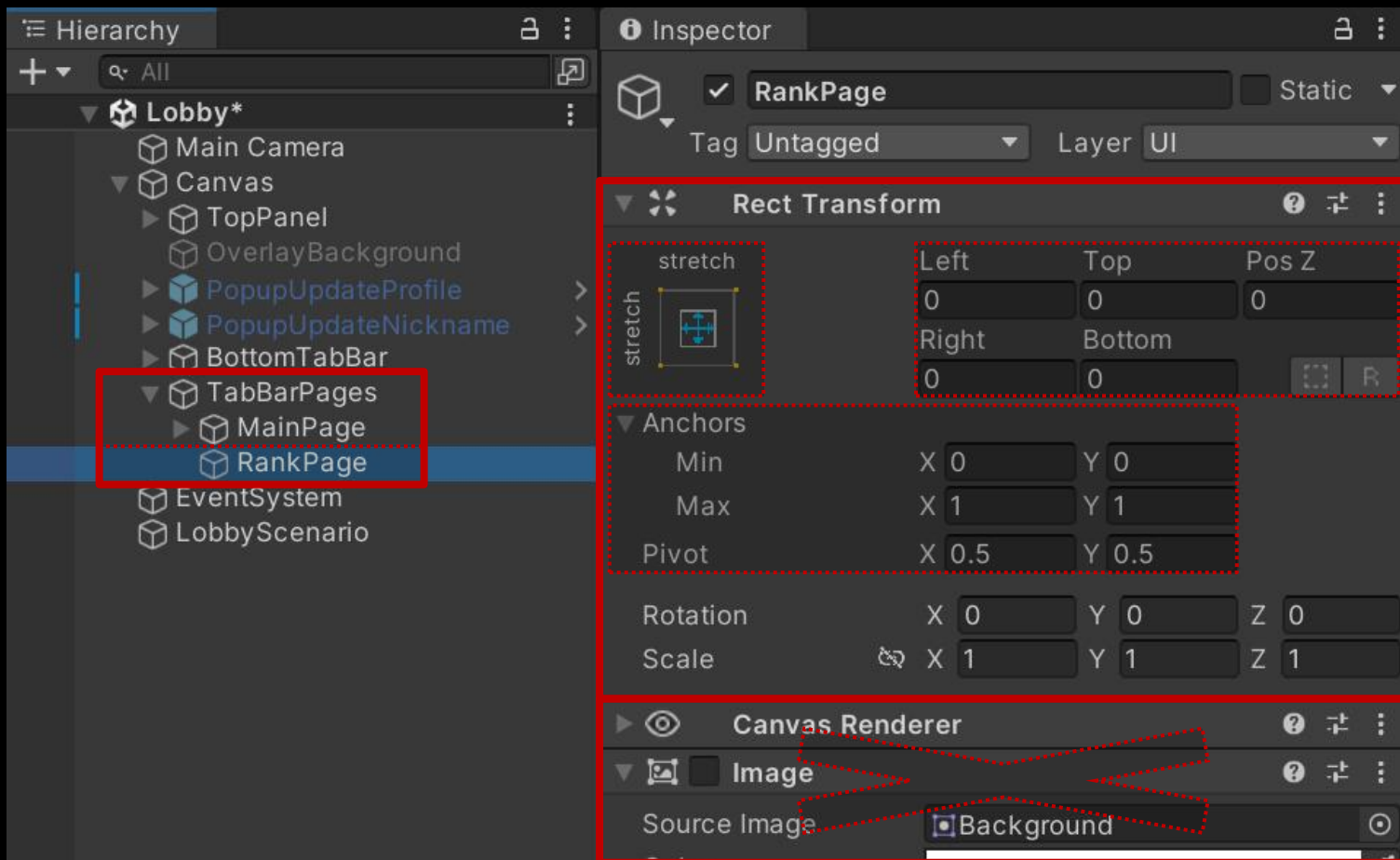
The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Main' object is selected and highlighted with a red box. In the Inspector panel, the 'Toggle' component is selected, and its 'On Value Changed (Boolean)' event is registered to 'GameObject.SetActive' on the 'MainPage' object. Red boxes and arrows highlight these specific elements.



# 랭킹 데이터 불러오기

## ■ Rank 탭 제작

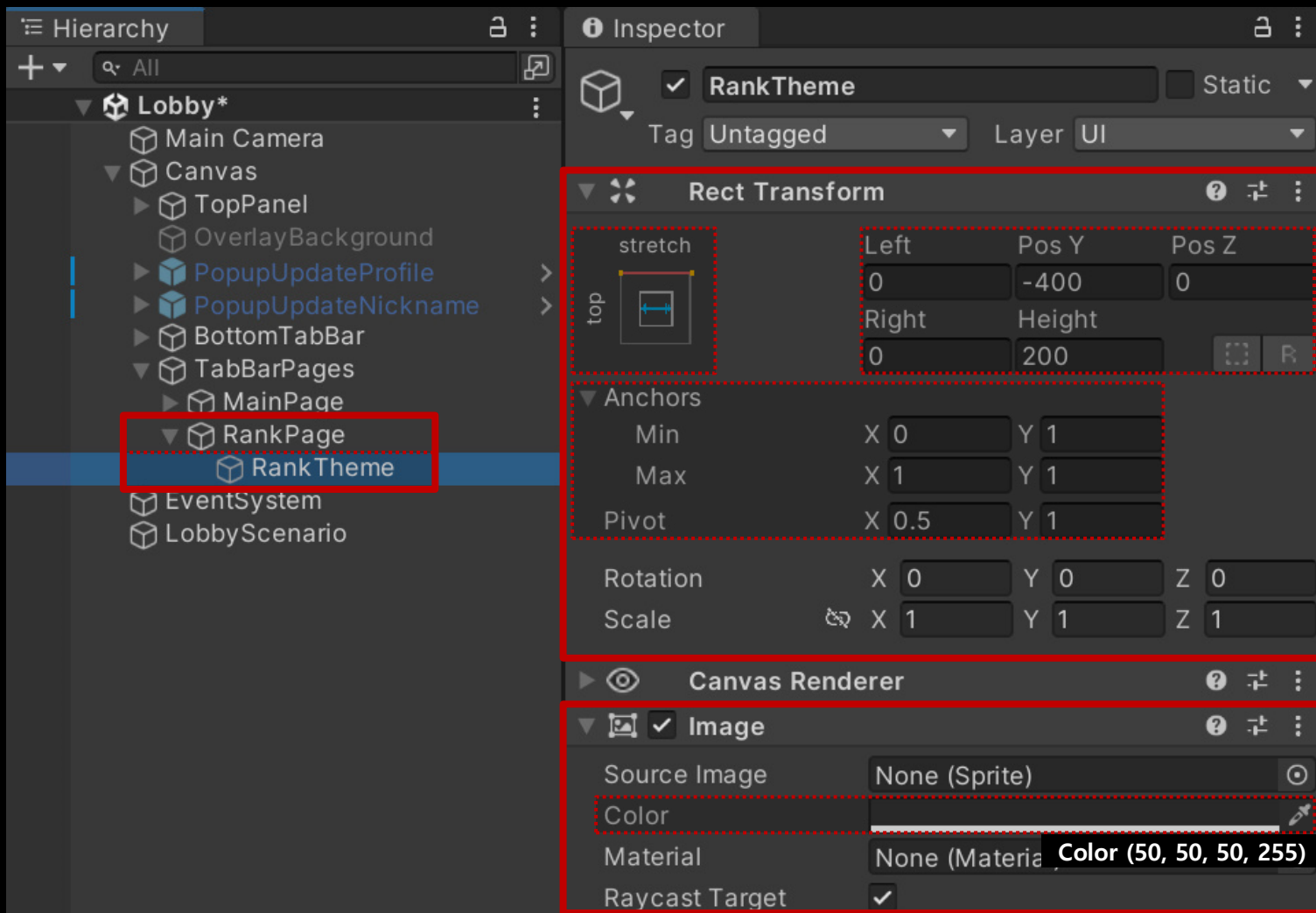
- 랭크 페이지를 관리하는 Panel UI 생성 및 설정
  - GameObject - UI - Panel





# 랭킹 데이터 불러오기

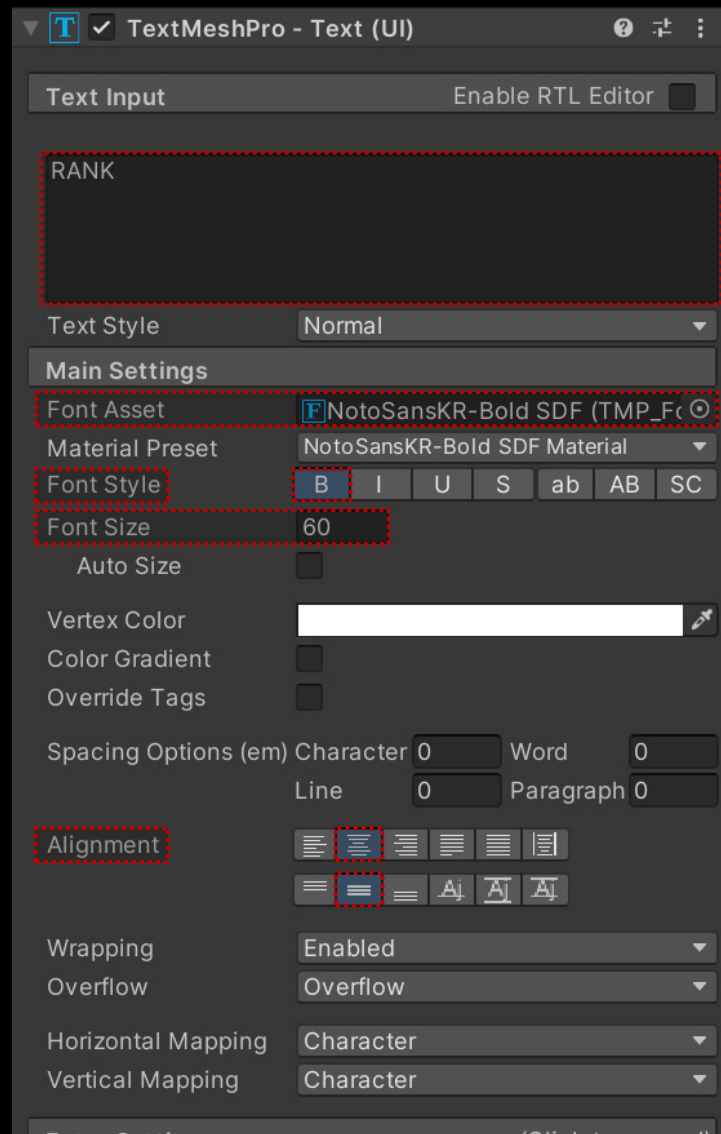
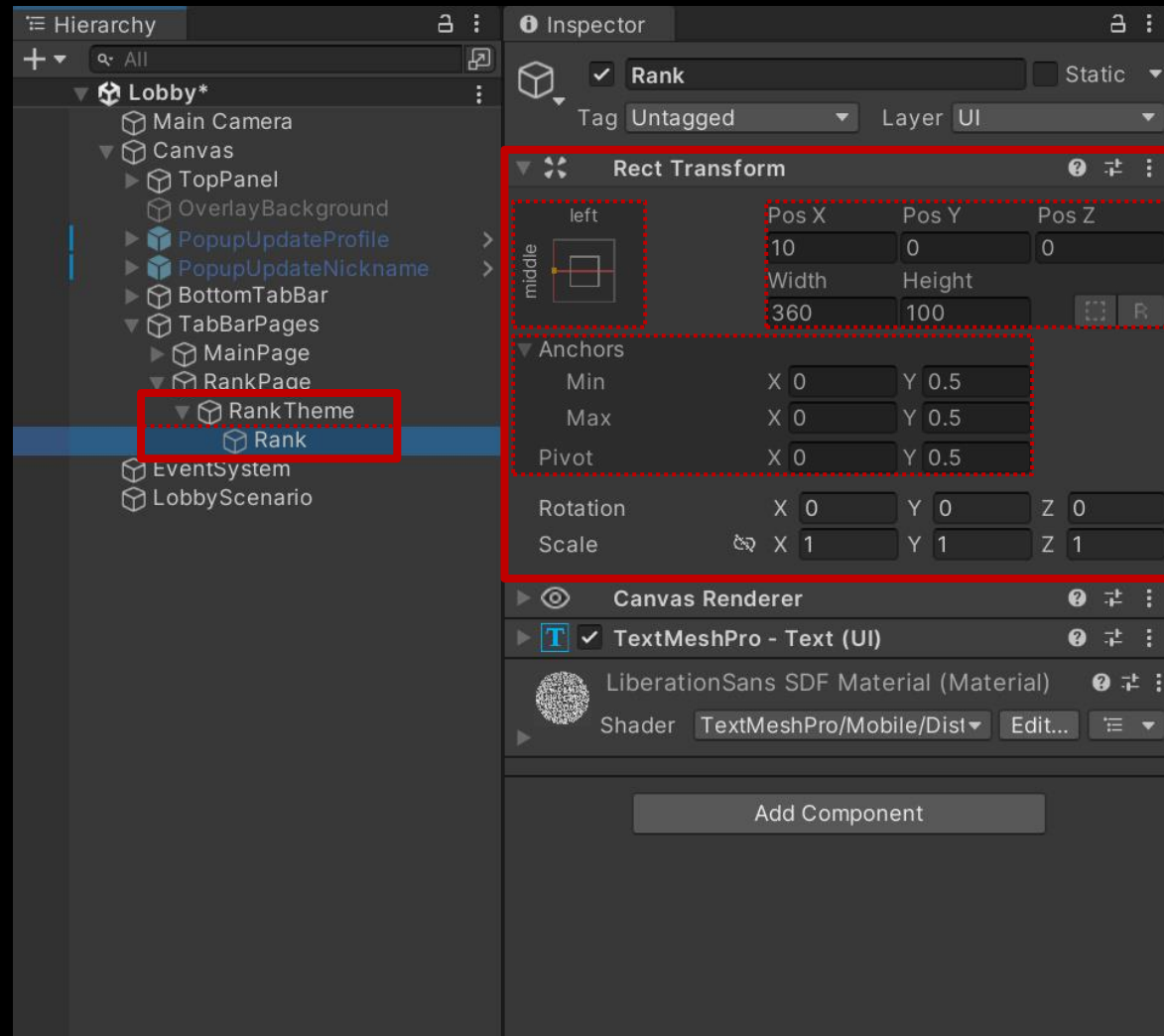
- 랭크 컬럼명의 뒷 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image





# 랭킹 데이터 불러오기

- “RANK” 텍스트를 출력하는 “Text - TextMeshPro” UI 생성 및 설정
  - GameObject - UI - “Text - TextMeshPro”







# 랭킹 데이터 불러오기

- "ICON" 텍스트를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - Rank 오브젝트를 Ctrl+D로 복제

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy, the 'Rank' object under 'RankPage' is selected, and its 'ProfileIcon' child is highlighted with a red box. The Inspector panel shows the 'ProfileIcon' component with the following settings:

- Tag: Untagged
- Layer: UI
- Static:

The 'Rect Transform' component is also visible, with the following settings:

- Position: Pos X: 380, Pos Y: 0, Pos Z: 0
- Size: Width: 180, Height: 100
- Anchor: left, middle
- Min: X 0, Y 0.5
- Max: X 0, Y 0.5
- Pivot: X 0, Y 0.5
- Rotation: X 0, Y 0, Z 0
- Scale: X 1, Y 1, Z 1

The 'Canvas Renderer' component is also visible, with the following settings:

- Text Input: ICON
- Enable RTL Editor:



# 랭킹 데이터 불러오기

- “NICKNAME” 텍스트를 출력하는 “Text - TextMeshPro” UI 생성 및 설정
  - Rank 오브젝트를 Ctrl+D로 복제

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy, the 'RankPage' folder is expanded, showing 'RankTheme', 'Rank', 'ProfileIcon', and 'Nickname'. The 'Nickname' object is selected. The Inspector panel shows the following settings:

- Nickname** (Static): Tag: Untagged, Layer: UI
- Rect Transform**:
  - Position: Pos X: -400, Pos Y: 0, Pos Z: 0
  - Size: Width: 480, Height: 100
  - Anchor: middle, right
  - 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**
- TextMeshPro - Text (UI)**:
  - Text Input: NICKNAME
  - Enable RTL Editor:



# 랭킹 데이터 불러오기

- "SCORE" 텍스트를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - Rank 오브젝트를 Ctrl+D로 복제

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby\*' with 'RankPage' expanded to show 'RankTheme', which contains 'Rank', 'ProfileIcon', 'Nickname', and 'Score'. The 'Score' object is selected. The Inspector panel on the right shows the properties for the selected 'Score' object, which is a 'TextMeshPro - Text (UI)'. The 'Rect Transform' component is expanded, showing 'Anchors' set to 'Min X 1, Y 0.5' and 'Max X 1, Y 0.5', and 'Pivot' set to 'X 1, Y 0.5'. The 'Text Input' field contains the text 'SCORE'. The 'Canvas Renderer' component is also visible below the 'TextMeshPro' component.



# 랭킹 데이터 불러오기

- 순위별 랭킹 정보를 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 'RankList' selected. The Inspector panel on the right shows the properties for the 'RankList' object, which is a 'Rect Transform' component. The 'Rect Transform' component is highlighted with a red dashed box, showing its position and anchors. The 'Scroll Rect' component is also highlighted with a red dashed box, showing its content and scroll settings.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
      - MainPage
        - RankPage
          - RankTheme
          - RankList (Selected)
          - Viewport
          - Scrollbar Horizontal
          - Scrollbar Vertical

- EventSystem
- LobbyScenario

**Inspector Panel:**

- RankList** (Static)
  - Tag: Untagged
  - Layer: UI
- Rect Transform**
  - stretch (checked)
  - Left: 0, Top: 600, Pos Z: 0
  - Right: 0, 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
- Canvas Renderer**
- Image** (checked)
- Scroll Rect**
  - Content: Content (Rect Transform)
  - Horizontal:
  - Vertical:



# 랭킹 데이터 불러오기

- Scrollbar가 보이지 않도록 Width를 0으로 설정

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

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
      - MainPage
      - RankPage
        - RankTheme
        - RankList
        - Viewport
        - Scrollbar Vertical** (highlighted with a red box)
  - EventSystem
  - LobbyScenario

**Inspector Panel:**

- Scrollbar Vertical (Static) [Tag: Untagged, Layer: UI]
- Rect Transform** (highlighted with a red box)
  - right: Pos X: 0, Top: 0, Pos Z: 0
  - stretch: Width: 0 (highlighted with a red dashed box), Bottom: 17
  - ▼ Anchors
    - 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
- Default UI Material (Material)
  - Shader: UI/Default

At the bottom of the Inspector panel, there is an "Add Component" button.



# 랭킹 데이터 불러오기

- Content 오브젝트에 컴포넌트 추가 및 설정

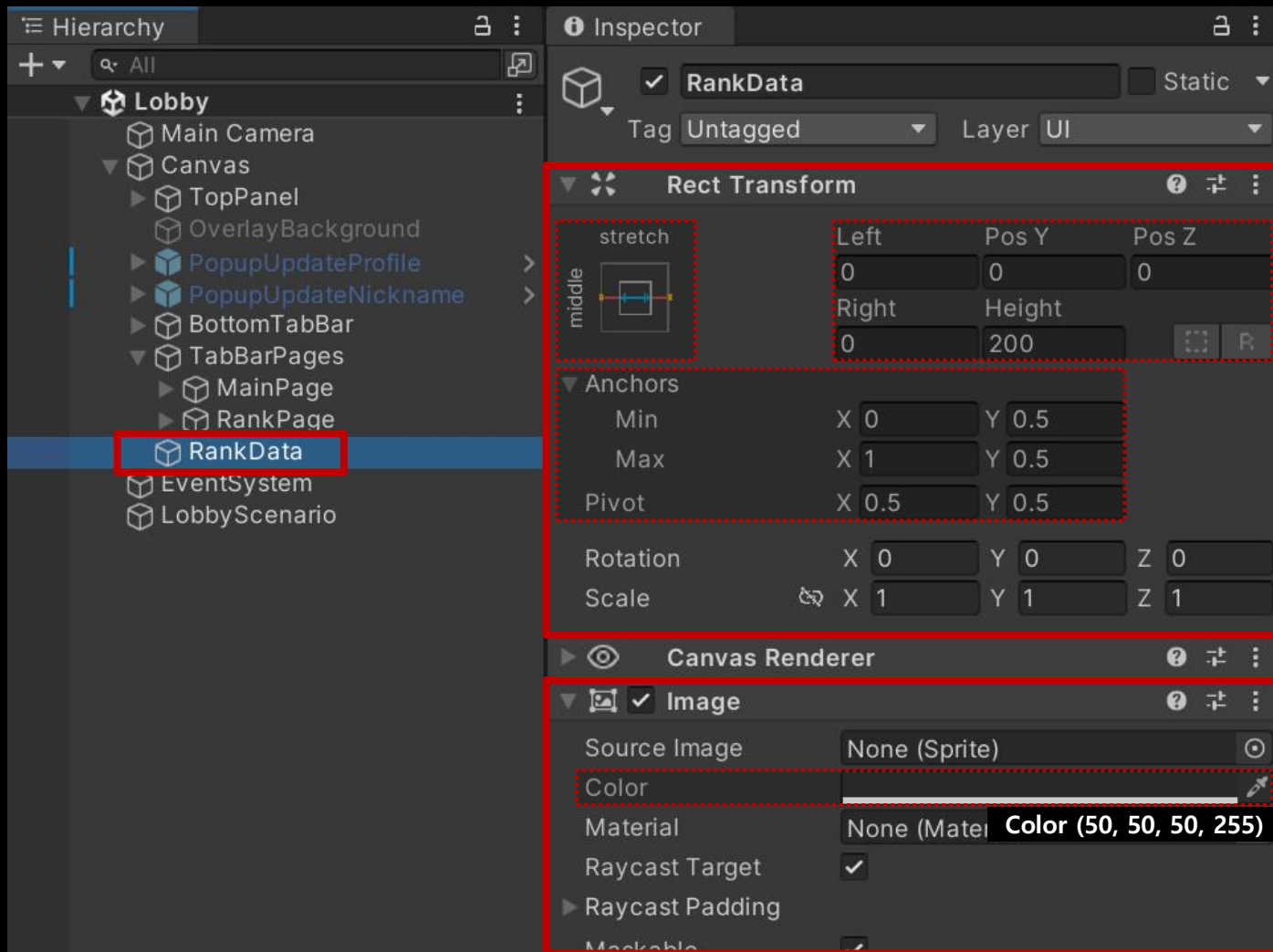
The screenshot displays the Unity Inspector for a **Content** object. The Hierarchy panel on the left shows the object's position within the scene hierarchy, with **Content** highlighted. The Inspector panel on the right shows the following components and their settings:

- Rect Transform**: stretch (middle), Left: 0, Pos Y: 0, Pos Z: 0, Right: 0, Height: 10.
- 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.
- Vertical Layout Group**: Padding (Left: 5, Right: 5, Top: 5, Bottom: 5), Spacing: 5, Child Alignment: Upper Center, Reverse Arrangement: , Control Child Size: Width , Height , Use Child Scale: Width , Height , Child Force Expand: Width , Height .
- Content Size Fitter**: Horizontal Fit: Unconstrained, Vertical Fit: Preferred Size.



# 랭킹 데이터 불러오기

- 유저 랭킹 정보를 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image





# 랭킹 데이터 불러오기

- 유저의 순위를 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 'Rank' object is selected under 'RankData'. The Inspector panel shows the 'Rank' object's properties, including 'Rect Transform' (Position: 10, 0, 0; Width: 360; Height: 100) and 'Anchors' (Min: X 0, Y 0.5; Max: X 0, Y 0.5; Pivot: X 0, Y 0.5). The 'TextMeshPro - Text (UI)' component is also visible.

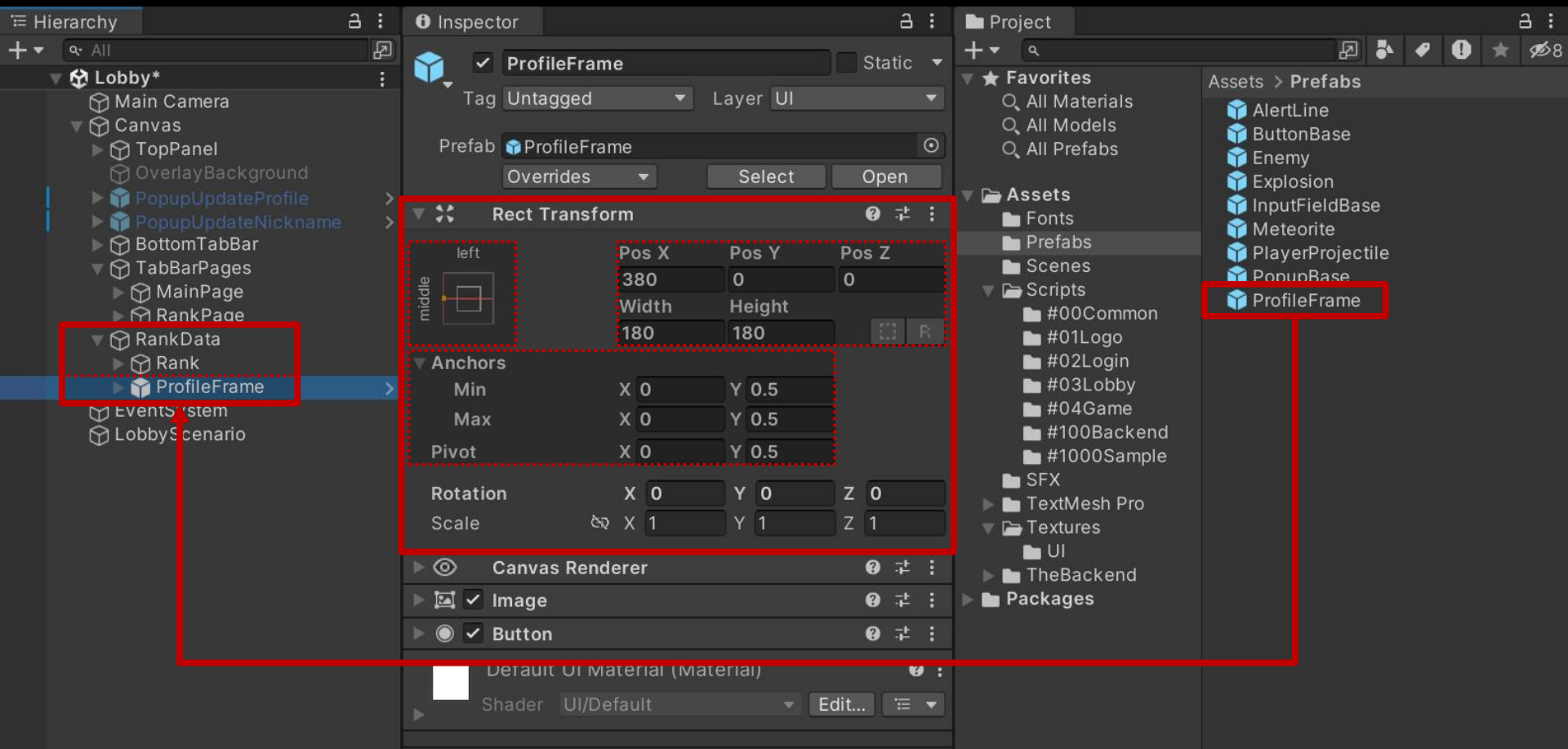
The screenshot shows the 'TextMeshPro - Text (UI)' component inspector. The 'Text Input' field contains '순위에 없음'. The 'Main Settings' section shows 'Font Asset' set to 'NotoSansKR-Bold SDF (TMP\_Fc)', 'Material Preset' set to 'NotoSansKR-Bold SDF Material', 'Font Style' set to 'B I U S ab AB SC', and 'Font Size' set to 60. The 'Alignment' section shows the 'Left' alignment icon selected.





# 랭킹 데이터 불러오기

- 유저의 아이콘을 출력하는 Image UI 생성 및 설정
  - ProfileFrame 프리팹을 Hierarchy View로 Drag & Drop





# 랭킹 데이터 불러오기

- 유저의 닉네임을 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - Rank 오브젝트를 Ctrl+D로 복제

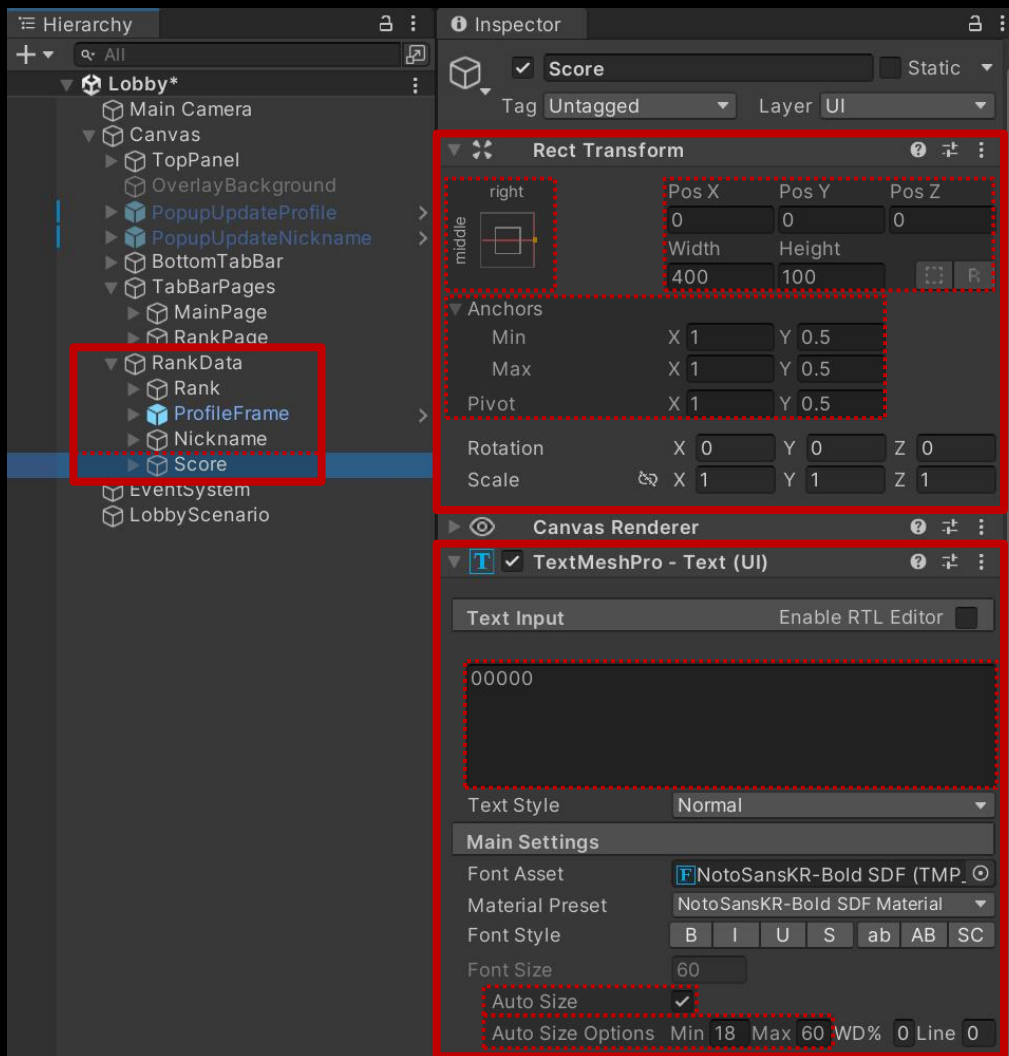
The screenshot shows the Unity Inspector window with the following configuration for the 'Nickname' UI element:

- Hierarchy:** Lobby\* > Canvas > RankData > Rank > ProfileFrame > Nickname
- Inspector:**
  - Nickname:** Tag: Untagged, Layer: UI
  - Rect Transform:**
    - Position: Pos X: -400, Pos Y: 0, Pos Z: 0
    - Size: Width: 480, Height: 100
    - Anchor: middle, right
    - 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:** TextMeshPro - Text (UI)
  - Text Input:** Enable RTL Editor:
  - Text:** 이름없음



# 랭킹 데이터 불러오기

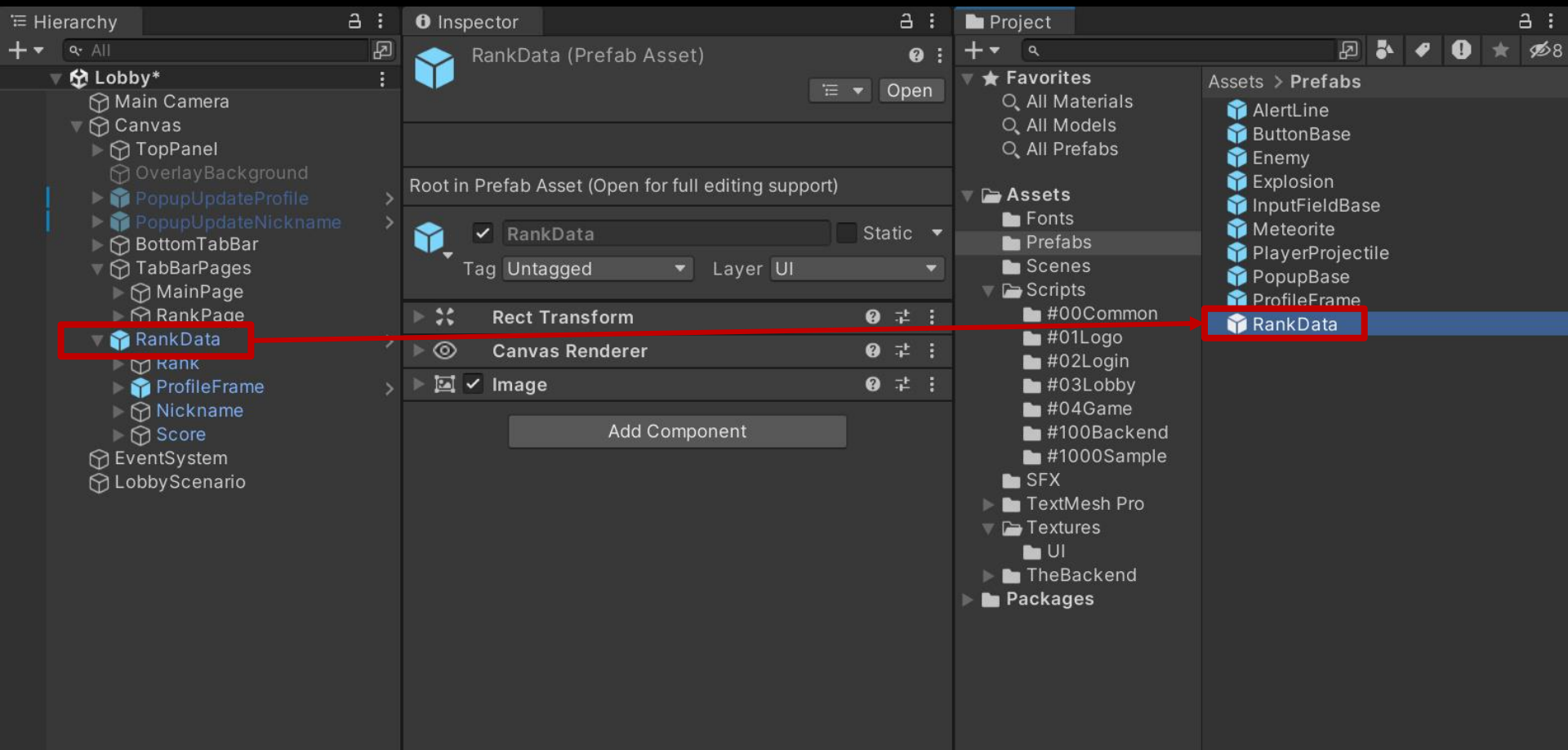
- 유저의 점수를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - Rank 오브젝트를 Ctrl+D로 복제





# 랭킹 데이터 불러오기

- RankData 오브젝트 Prefab 생성
  - Hierarchy View의 "RankData" 오브젝트를 Project View로 Drag & Drop





# 랭킹 데이터 불러오기

## ■ RankData 오브젝트 설정

The screenshot shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'RankPage' object is expanded, and 'MyRankData' is selected. In the Inspector panel, the 'MyRankData' object is selected, and its 'Rect Transform' component is highlighted with a red dashed box. The 'Rect Transform' component shows the following settings:

Property	Value
Left	0
Right	0
Pos Y	400
Height	200
Pos Z	0
Min X	0
Max X	1
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

The 'Anchors' section is also visible, showing 'Min' (X 0, Y 0) and 'Max' (X 1, Y 0) settings. The 'Pivot' is set to (0.5, 0). The 'Rotation' and 'Scale' sections show default values (0, 0, 0 for rotation and 1, 1, 1 for scale).

1. RankData 오브젝트의 이름을 MyRankData로 변경
2. RankPage 오브젝트의 자식으로 설정



# 랭킹 데이터 불러오기

## ■ 랭킹 데이터 불러오기

- 랭킹을 출력할 최대 인원 수 설정
  - Constants Script 수정

```
1 public static class Constants
2 {
3     public static readonly string USER_DATA_TABLE = "USER_DATA";
4     public static readonly string DAILY_RANK_UUID = "e795f580-04f4-11ee-958d-ad4ae0c79832";
5     public static readonly int MAX_RANK_LIST = 20;
6 }
```



# 랭킹 데이터 불러오기

- 개별 랭킹 정보의 UI 출력을 제어하는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "DailyRankData"로 변경

```
1 using UnityEngine;
2 using TMPro;
3
4 public class DailyRankData : MonoBehaviour
5 {
6     [SerializeField]
7     private TextMeshProUGUI textRank;
8     [SerializeField]
9     private TextMeshProUGUI textNickName;
10    [SerializeField]
11    private TextMeshProUGUI textScore;
12
13    private int    rank;
14    private string nickname;
15    private int    score;
16
```



# 랭킹 데이터 불러오기

- 개별 랭킹 정보의 UI 출력을 제어하는 스크립트 생성 및 작성 (계속)

```
17 public int Rank
18 {
19     set
20     {
21         if ( value <= Constants.MAX_RANK_LIST )
22         {
23             rank = value;
24             textRank.text = rank.ToString();
25         }
26         else
27         {
28             textRank.text = "순위에 없음";
29         }
30     }
31     get => rank;
32 }
33
```





# 랭킹 데이터 불러오기

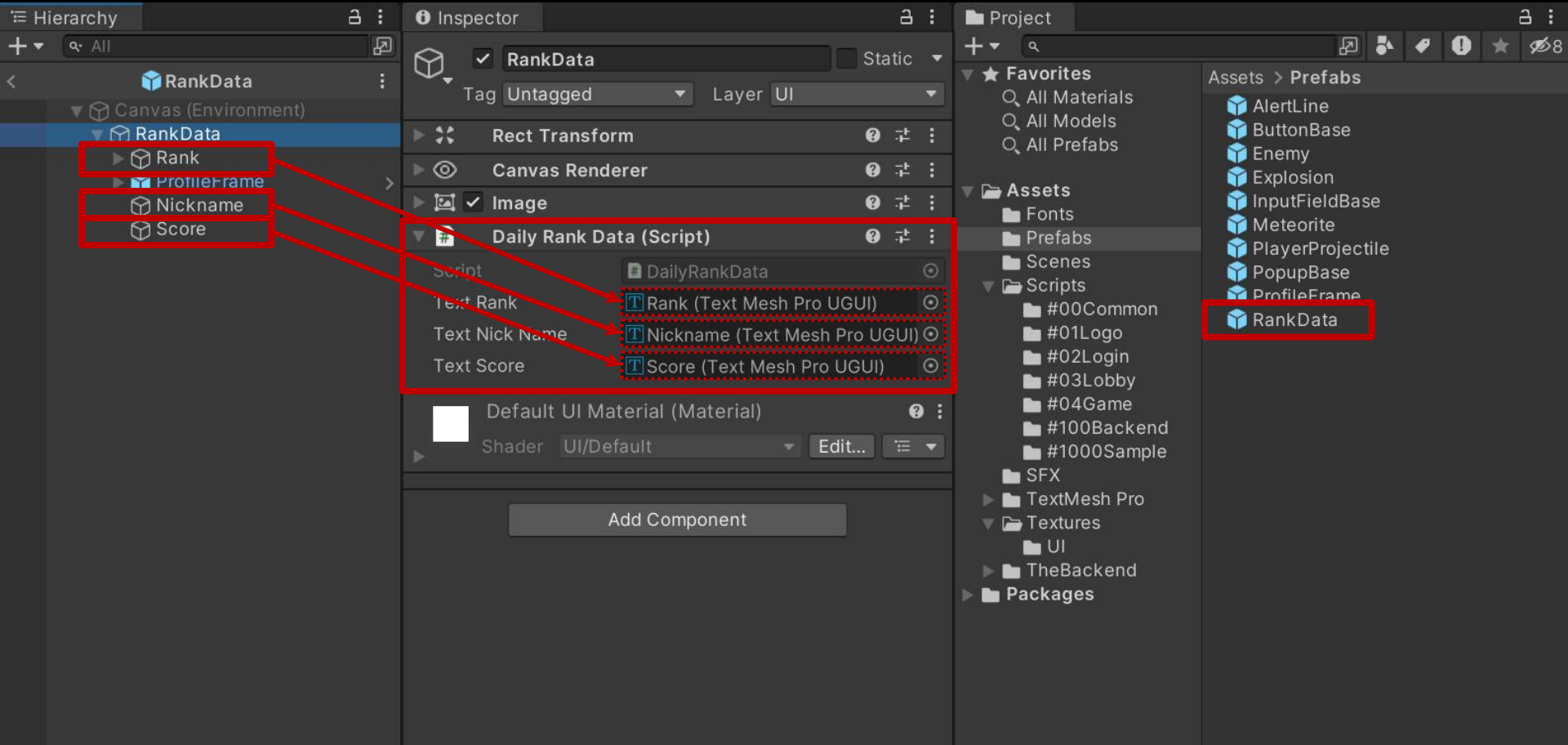
- 개별 랭킹 정보의 UI 출력을 제어하는 스크립트 생성 및 작성 (계속)

```
34 public string Nickname
35 {
36     set
37     {
38         nickname = value;
39         textNickName.text = nickname;
40     }
41     get => nickname;
42 }
43
44 public int Score
45 {
46     set
47     {
48         score = value;
49         textScore.text = score.ToString();
50     }
51     get => score;
52 }
53 }
```



# 랭킹 데이터 불러오기

- RankData 프리팹에 DailyRankData 컴포넌트 추가 및 설정





# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "DailyRankLoader"로 변경

```
1 using System.Collections.Generic;
2 using UnityEngine;
3 using UnityEngine.UI;
4 using BackEnd;
5
6 public class DailyRankLoader : MonoBehaviour
7 {
8     [SerializeField]
9     private GameObject rankDataPrefab; // 랭킹 정보 출력을 위한 UI 프리팹 원본
10    [SerializeField]
11    private Scrollbar scrollbar; // scrollbar의 value 설정 (활성화될 때 1위가 보이도록)
12    [SerializeField]
13    private Transform rankDataParent; // ScrollView의 Content 오브젝트
14    [SerializeField]
15    private DailyRankData myRankData; // 내 랭킹 정보를 출력하는 UI 게임오브젝트
16
17    private List<DailyRankData> rankDataList;
18
```



# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
19 private void Awake()
20 {
21     rankDataList = new List<DailyRankData>();
22
23     // 1 ~ 20위 랭킹 출력을 위한 UI 오브젝트 생성
24     for ( int i = 0; i < Constants.MAX_RANK_LIST; ++ i )
25     {
26         GameObject clone = Instantiate(rankDataPrefab, rankDataParent);
27         rankDataList.Add(clone.GetComponent<DailyRankData>());
28     }
29 }
30
31 private void OnEnable()
32 {
33     // 1위 랭킹이 보이도록 scrollbar 값 설정
34     scrollbar.value = 1;
35     // 1 ~ 20위의 랭킹 정보 불러오기
36     GetRankList();
37     // 내 랭킹 정보 불러오기
38     GetMyRank();
39 }
40
```



# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

== backend method ==

Backend.URank.User.GetRankList(string rankUuid, int limit=10 int offset=0);

rankUuid 랭킹 테이블에 있는 유저의 offset ~ offset + limit 순위 랭킹 정보를 불러온다.

```
41 private void GetRankList()
42 {
43     // 1 ~ 20위 랭킹 정보 불러오기
44     Backend.URank.User.GetRankList(Constants.DAILY_RANK_UUID, Constants.MAX_RANK_LIST, callback =>
45     {
46         if (callback.IsSuccess())
47         {
48             // JSON 데이터 파싱 성공
49             try
50             {
51                 Debug.Log($"랭킹 조회에 성공했습니다 : {callback}");
52
53                 LitJson.JsonData rankDataJson = callback.FlattenRows();
54
55                 // 받은 데이터의 개수가 0이면 데이터가 없는 것
56                 if (rankDataJson.Count <= 0)
57                 {
58                     // 1 ~ 20위까지 데이터를 빈 데이터로 설정
59                     for (int i = 0; i < Constants.MAX_RANK_LIST; ++i)
60                     {
61                         SetRankData(rankDataList[i], i + 1, "-", 0);
62                     }
63
64                     Debug.LogWarning("데이터가 존재하지 않습니다.");
65                 }
66             }
67             catch { }
68         }
69     }
70 }
```

Tip. limit 매개변수에는 1~100까지 숫자 입력 가능



# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
66     else
67     {
68         int rankerCount = rankDataJson.Count;
69
70         // 랭킹 정보를 불러와 출력할 수 있도록 설정
71         for (int i = 0; i < rankerCount; ++i)
72         {
73             rankDataList[i].Rank    = int.Parse(rankDataJson[i]["rank"].ToString());
74             rankDataList[i].Score    = int.Parse(rankDataJson[i]["score"].ToString());
75
76             // 닉네임은 별도로 설정하지 않은 유저도 존재할 수 있기 때문에
77             // 닉네임이 존재하지 않는 유저는 닉네임 대신 gameId를 출력
78             rankDataList[i].Nickname = rankDataJson[i].ContainsKey("nickname") == true ?
79                                     rankDataJson[i]["nickname"].ToString() : UserInfo.Data.gamerId;
80         }
81         // 만약 limitCount에 설정된 숫자보다 현재 랭킹에 등록된 숫자가 적으면 나머지는 빈 값으로 설정
82         for (int i = rankerCount; i < Constants.MAX_RANK_LIST; ++i)
83         {
84             SetRankData(rankDataList[i], i + 1, "-", 0);
85         }
86     }
87 }
```



# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
88 // JSON 데이터 파싱 실패
89 catch (System.Exception e)
90 {
91     // try-catch 에러 출력
92     Debug.LogError(e);
93 }
94 }
95 else
96 {
97     // 1 ~ 20위까지 데이터를 빈 데이터로 설정
98     for (int i = 0; i < Constants.MAX_RANK_LIST; ++i)
99     {
100         SetRankData(rankDataList[i], i + 1, "-", 0);
101     }
102
103     Debug.LogError($"랭킹 조회 중 오류가 발생했습니다 : {callback}");
104 }
105 });
106 }
107
```



# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
108 private void GetMyRank()
109 {
110     // 내 랭킹 정보 불러오기
111     Backend.URank.User.GetMyRank(Constants.DAILY_RANK_UUID, callback =>
112     {
113         // 닉네임이 없으면 gameId, 닉네임이 있으면 nickname 사용
114         string nickname = UserInfo.Data.nickname == null ? UserInfo.Data.gameId : UserInfo.Data.nickname;
115
116         if ( callback.IsSuccess() )
117         {
118             // JSON 데이터 파싱 성공
119             try
120             {
121                 LitJson.JsonData rankDataJson = callback.FlattenRows();
122
123                 // 받아온 데이터의 개수가 0이면 데이터가 없는 것
124                 if ( rankDataJson.Count <= 0 )
125                 {
126                     // ["순위에 없음", "닉네임", 0]과 같이 출력
127                     SetRankData(myRankData, 1000000000, nickname, 0);
128
129                     Debug.LogWarning("데이터가 존재하지 않습니다.");
130                 }
131             }
132         }
133     }
134 }
```





# 랭킹 데이터 불러오기

- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
131     else
132     {
133         myRankData.Rank      = int.Parse(rankDataJson[0]["rank"].ToString());
134         myRankData.Score    = int.Parse(rankDataJson[0]["score"].ToString());
135
136         // 닉네임은 별도로 설정하지 않은 유저도 존재할 수 있기 때문에
137         // 닉네임이 존재하지 않는 유저는 닉네임 대신 gameId를 출력
138         myRankData.Nickname = rankDataJson[0].ContainsKey("nickname") == true ?
139                               rankDataJson[0]["nickname"].ToString() : UserInfo.Data.gamerId;
140     }
141 }
142 // 자신의 랭킹 정보 JSON 데이터 파싱에 실패했을 때
143 catch ( System.Exception e )
144 {
145     // ["순위에 없음", "닉네임", 0]과 같이 출력
146     SetRankData(myRankData, 1000000000, nickname, 0);
147
148     // try-catch 에러 출력
149     Debug.LogError(e);
150 }
151 }
```



# 랭킹 데이터 불러오기

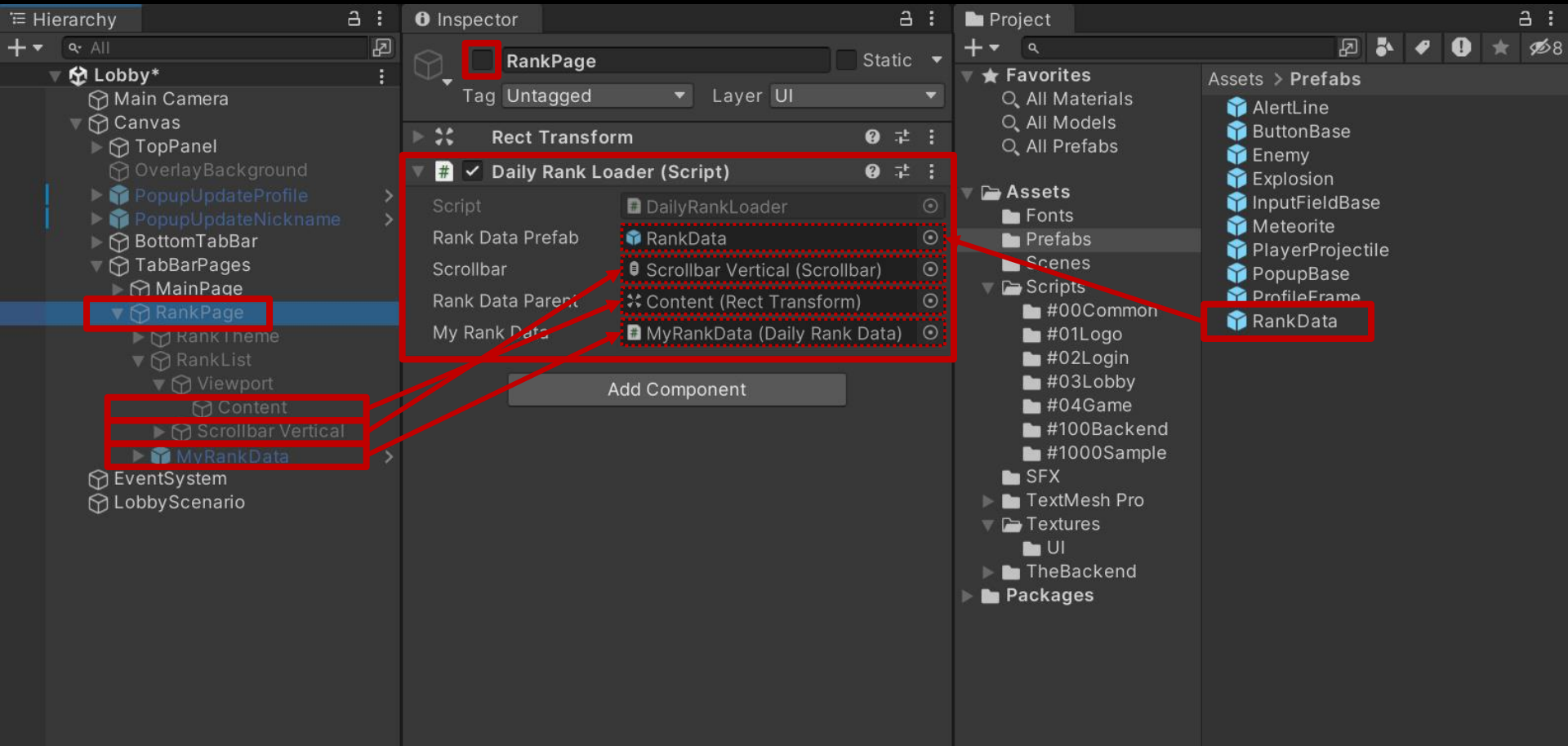
- 일일 최고 점수 랭킹을 불러오는 스크립트 생성 및 작성 (계속)

```
152     else
153     {
154         // 자신의 랭킹 정보 데이터가 존재하지 않을 때
155         if ( callback.GetMessage().Contains("userRank") )
156         {
157             // ["순위에 없음", "닉네임", 0]과 같이 출력
158             SetRankData(myRankData, 100000000, nickname, 0);
159         }
160     }
161 });
162 }
163
164 private void SetRankData(DailyRankData rankData, int rank, string nickname, int score)
165 {
166     rankData.Rank        = rank;
167     rankData.Nickname    = nickname;
168     rankData.Score       = score;
169 }
170 }
```



# 랭킹 데이터 불러오기

- RankPage 오브젝트에 "DailyRankLoader" 컴포넌트 추가 및 설정





# 랭킹 데이터 불러오기

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

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Ranking' object is selected under the 'Lobby\*' folder. The Inspector panel shows the 'Toggle' component selected. The 'On Value Changed (Boolean)' event is configured with the following settings:

- Runtime Only:
- Event Name: `GameObject.SetActive`
- Target: `RankPage`

Red boxes and arrows highlight the 'Ranking' object in the Hierarchy, the 'Toggle' component in the Inspector, and the 'RankPage' target in the event list.



# 랭킹 데이터 불러오기

## ■ 결과 화면

