

# Unity Backend

“우편” 기능을 이용한 유저 우편 관리 (UI)

Created in 2023-06-21

Last Updated 2023-06-21

Unity Version 2022.2.2f1

## *Index*

- ◆ **우편함 (Post Box)**
- ◆ **우편 (Post)**
- ◆ **우편 데이터와 UI 연동**

# 우편함 (Post Box)

- 우편함



# 우편함 (Post Box)

## ■ 우편함

- 우편함 열기 "Button - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

The screenshot displays the Unity development environment with the following components highlighted:

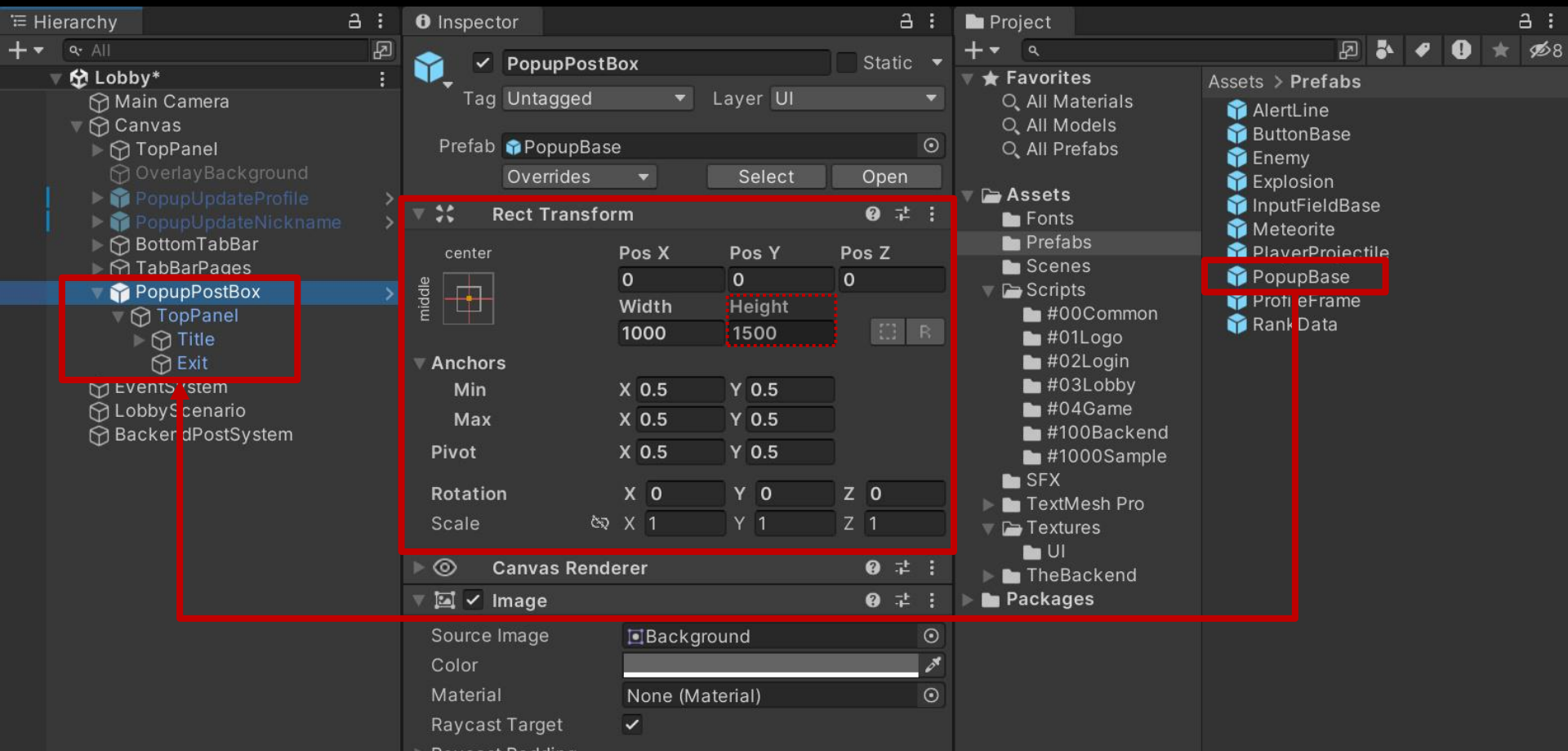
- Hierarchy Panel:** Shows the scene structure under 'Lobby\*'. The path 'MainPage > ButtonPost' is highlighted with a red box.
- Inspector Panel:** Shows the properties for the selected 'ButtonPost' object.
  - Rect Transform:** A red dashed box highlights the 'left' and 'top' anchors, and the 'Pos X' (50), 'Pos Y' (-250), 'Width' (150), and 'Height' (150) fields.
  - Canvas Renderer:** Shows the 'Image' component with 'Source Image' set to 'UI\_Button\_Post'.
- Project Panel:** Shows the 'Assets > Textures > UI' folder structure. The file 'UI\_Button\_Post' is highlighted with a red box and a red arrow pointing to the 'Source Image' field in the Inspector.

**“UI\_Button\_Post.png” 이미지가 없을 경우  
[뒤끝 강의] 온라인 게임 개발 [11][Resources] 다운로드**



# 우편함 (Post Box)

- 우편을 관리하는 우편함 UI 생성 및 설정
  - PopupBase 프리팹을 Hierarchy View로 Drag & Drop





# 우편함 (Post Box)

- Title 오브젝트의 "TextMeshPro - Text" 컴포넌트 변수 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Title' object under 'Lobby\*' is selected and highlighted with a red box. The Inspector panel shows the 'TextMeshPro - Text (UI)' component selected, also highlighted with a red box. The 'Text Input' field contains the Korean text '우편함'. The 'Main Settings' section is visible, showing the following configurations:

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



# 우편함 (Post Box)

- 우편 리스트를 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 'PostList' selected under 'PopupPostBox'. The Inspector panel on the right shows the properties for the 'PostList' object, which is a 'Rect Transform' component. The 'Rect Transform' component is highlighted with a red dashed box, showing its position and anchors. The 'Canvas Renderer' component is also highlighted with a red dashed box. The 'Scroll Rect' component is highlighted with a red dashed box, showing its content and scroll settings.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - TopPanel
      - PostList (Selected)
      - Viewport
        - Content
        - Scrollbar Horizontal
        - Scrollbar Vertical
  - EventSystem
  - LobbyScenario
  - BackendPostSystem

**Inspector Panel:**

- PostList** (Static)
  - Tag: Untagged
  - Layer: UI
- Rect Transform**
  - stretch (Left, Top, Right, Bottom)
  - Left: 10, Top: 110, Right: 10, Bottom: 120
  - 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:
  - Vertical:



# 우편함 (Post Box)

- Scrollbar가 보이지 않도록 Width를 0으로 설정

The screenshot shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Scrollbar Vertical' component is selected under 'Lobby\*' > 'PopupPostBox' > 'PostList'. The Inspector panel shows the 'Scrollbar Vertical' component with the following properties:

- Tag: Untagged
- Layer: UI
- Rect Transform:
  - right: Pos X 0, Top 0, Pos Z 0
  - stretch: Width 0, 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

The 'Width' property in the Rect Transform section is highlighted with a red dashed box, indicating it is set to 0.





# 우편함 (Post Box)

- Content 오브젝트에 컴포넌트 추가 및 설정

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

- Content** component: Tag: Untagged, Layer: UI, Static:
- Rect Transform** component: stretch: Left: 0, Pos Y: 0, Pos Z: 0, Right: 0, Height: 10
- Anchors** component: Min: X: 0, Y: 1; Max: X: 1, Y: 1; Pivot: X: 0, Y: 1; Rotation: X: 0, Y: 0, Z: 0; Scale: X: 1, Y: 1, Z: 1
- Vertical Layout Group** component (highlighted with a red dashed border):
  - Padding: Left: 5, Right: 5, Top: 5, Bottom: 5
  - 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** component (highlighted with a red dashed border):
  - Horizontal Fit: Unconstrained
  - Vertical Fit: Preferred Size



# 우편함 (Post Box)

- 우편 전체 수령 "Button - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the **ButtonReceiveAll** object under **PopupPostBox** highlighted in a red box. The Inspector panel on the right shows the properties of the selected **ButtonReceiveAll** object, with several sections highlighted in red dashed boxes:

- Rect Transform**: Shows the object's position and size. The **right** and **bottom** anchors are visible. The **Pos X** is -10, **Pos Y** is 10, and **Pos Z** is 0. The **Width** is 200 and **Height** is 100.
- Anchors**: Shows the **Min** and **Max** anchor settings, both set to X 1 and Y 0. The **Pivot** is also set to X 1 and Y 0.
- Canvas Renderer**: Shows the **Image** component with **Source Image** set to **UISprite**.
- Color**: The **Color** property is set to **Color (73, 105, 139, 255)**.



# 우편함 (Post Box)

- 우편 전체 수령 "Button - TextMeshPro" UI 생성 및 설정 (계속)

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby\*' with various UI elements. The 'Text (TMP)' element is highlighted with a red box. The Inspector panel on the right shows the properties for the selected 'TextMeshPro - Text (UI)' element, which is also highlighted with a red box. The Inspector panel includes a 'Text Input' field with the text '전체수령', 'Main Settings' section with 'Font Asset' set to 'NotoSansKR-Bold SDF (TMP)', 'Font Size' set to 30, and 'Vertex Color' set to 'Color (255, 255, 255, 255)'. The 'Text Input' field is also highlighted with a red dashed border.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - TopPanel
      - PostList
      - ButtonReceiveAll
- EventSystem
- LobbyScenario
- BackendPostSystem

**Inspector Panel:**

- Text (TMP) [Static]
- Tag: Untagged, Layer: UI
- Rect Transform
- Canvas Renderer
- TextMeshPro - Text (UI)**
  - Text Input: 전체수령 [Enable RTL Editor]
  - Text Style: Normal
  - Main Settings
    - Font Asset: NotoSansKR-Bold SDF (TMP)
    - Material Preset: NotoSansKR-BoId SDF Material
    - Font Style: B I U S ab AB SC
    - Font Size: 30 [Auto Size]
    - Vertex Color: Color (255, 255, 255, 255)
    - Color Gradient: [Color (255, 255, 255, 255)]



# 우편함 (Post Box)

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

Hierarchy

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - TopPanel
      - PostList
      - ButtonReceiveAll
      - TextSystem
    - EventSystem
    - LobbyScenario
    - BackendPostSystem

Inspector

TextSystem

Tag Untagged Layer UI

Rect Transform

stretch	Left	Pos Y	Pos Z
middle	10	0	0
	Right	Height	
	10	50	

Anchors

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

Rotation

X	Y	Z
0	0	0

Scale

X	Y	Z
1	1	1

Canvas Renderer

TextMeshPro - Text (UI)

NotoSansKR-Bold SDF Material (Material)

Shader TextMeshPro/Distance Fi

Add Component

TextMeshPro - Text (UI)

Text Input Enable RTL Editor

우편함이 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 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



# 우편함 (Post Box)

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

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure with 'Lobby\*' as the root, containing various UI components like 'Main Camera', 'Canvas', 'TopPanel', 'OverlayBackground', 'PopupUpdateProfile', 'PopupUpdateNickname', 'BottomTabBar', 'TabBarPages', 'MainPage', 'ButtonGameStart', 'ButtonPost', 'RankPage', 'PopupPostBox', 'EventSystem', 'LobbyScenario', and 'BackendPostSystem'. The Inspector panel on the right shows the 'ButtonPost' component selected, with its 'Image' sub-component active. The 'Button' component's 'onClick()' list contains four entries, each with a 'Runtime Only' dropdown, a method name, and a target object. The entries are: 1. 'Runtime Only' dropdown, 'RectTransform.SetAsLastSibling' method, target 'OverlayBa'. 2. 'Runtime Only' dropdown, 'GameObject.SetActive' method, target 'OverlayBa' with a checked checkbox. 3. 'Runtime Only' dropdown, 'RectTransform.SetAsLastSibling' method, target 'PopupPost'. 4. 'Runtime Only' dropdown, 'GameObject.SetActive' method, target 'PopupPost' with a checked checkbox. Red dashed boxes highlight these four entries. Red lines connect the 'OverlayBackground' and 'PopupPostBox' objects in the Hierarchy to their respective targets in the Inspector's event list.



# 우편함 (Post Box)

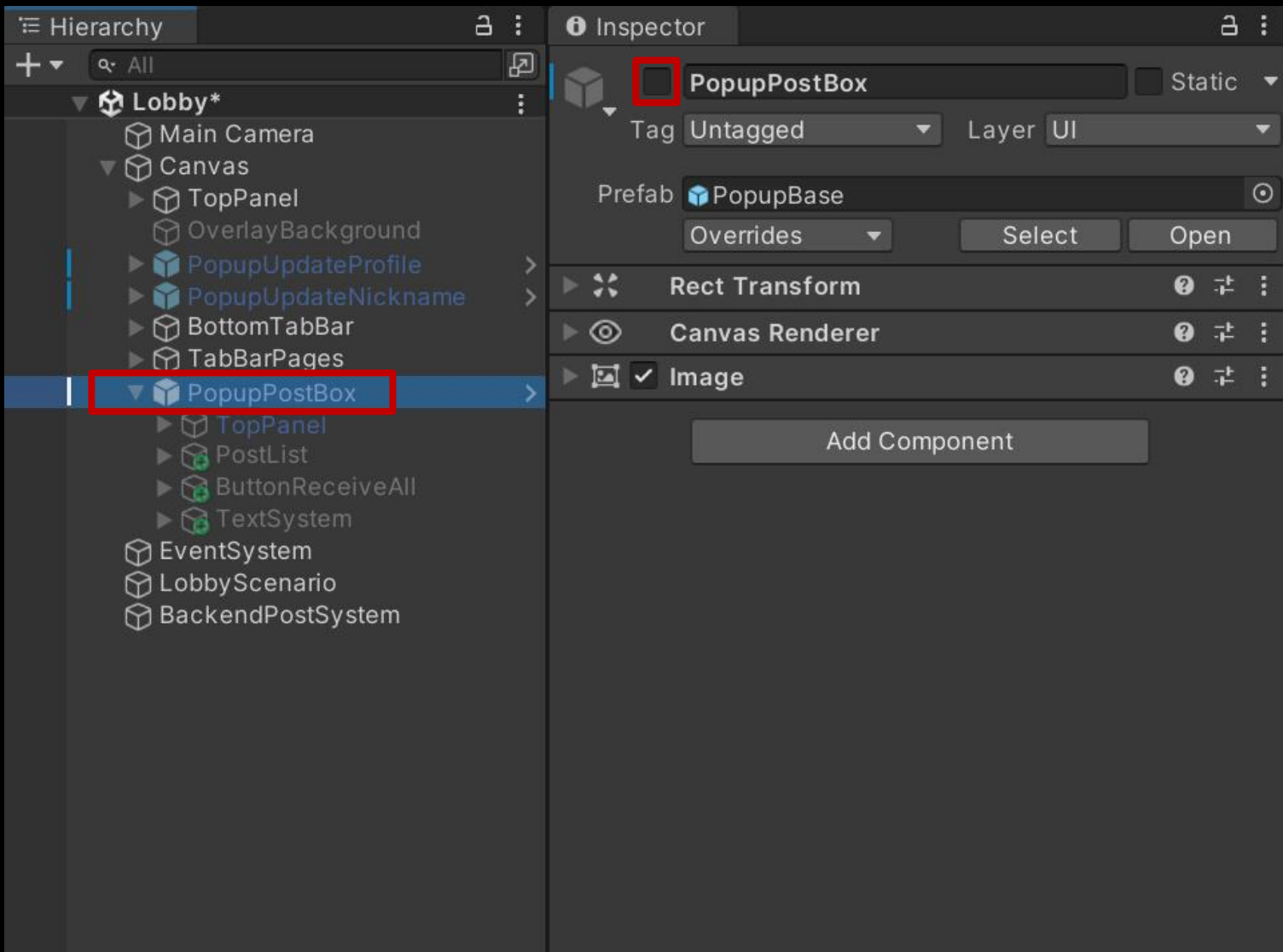
- Exit 오브젝트의 "Button" 컴포넌트 onClick() 이벤트 등록

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the 'Exit' object selected under 'PopupPostBox'. The Inspector panel on the right shows the properties of the 'Exit' object, including the 'Button' component. The 'onClick()' event list is expanded, showing two entries: 'Runtime Only' with 'GameObject.SetActive' and target 'OverlayBa', and another 'Runtime Only' with 'GameObject.SetActive' and target 'PopupPos'. Red boxes and arrows highlight the 'Exit' object in the Hierarchy, the 'Button' component in the Inspector, and the two event entries in the 'onClick()' list.



# 우편함 (Post Box)

- PopupPostBox 오브젝트 비활성화

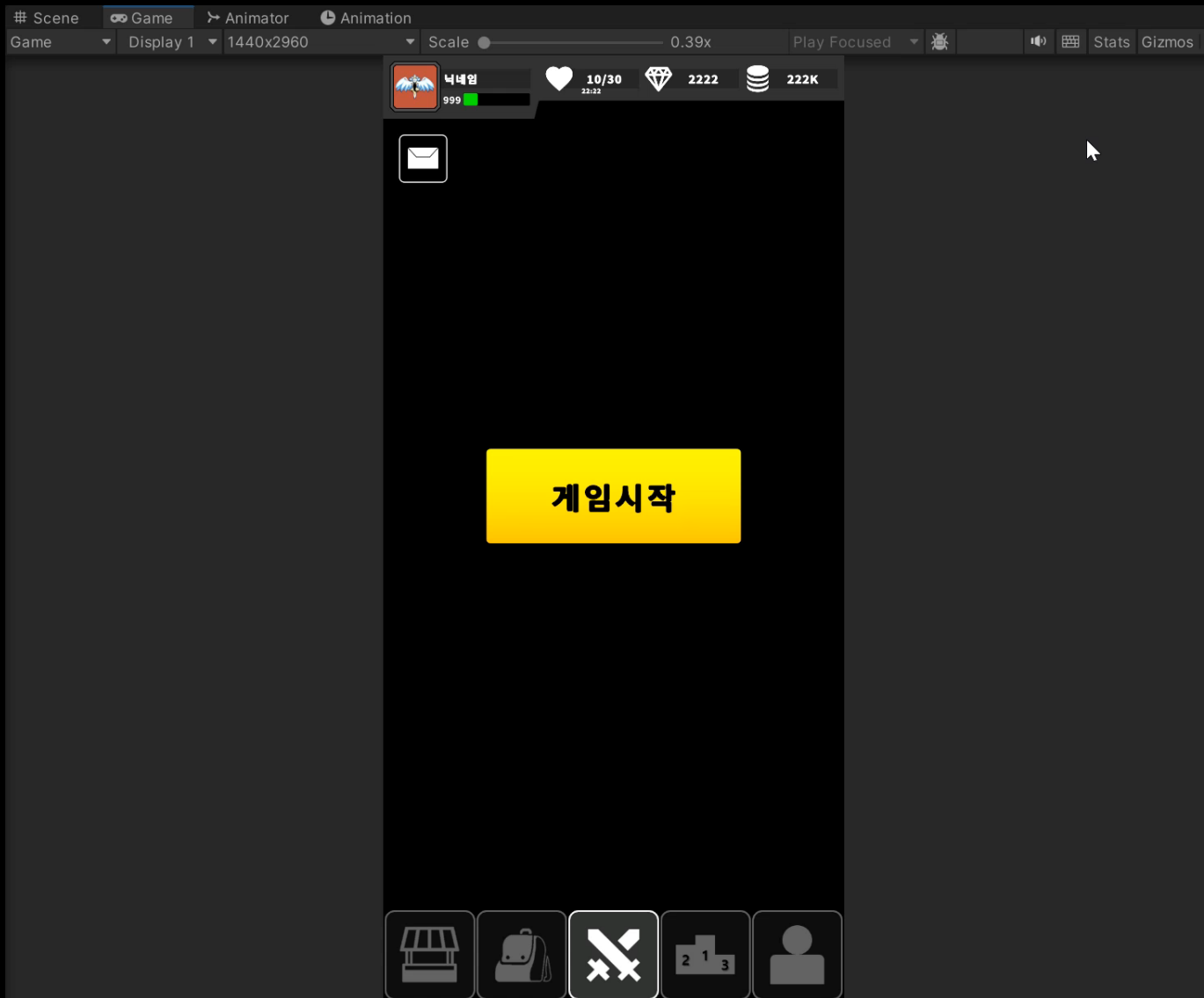






# 우편함 (Post Box)

## ■ 결과 화면





# 우편 (Post)

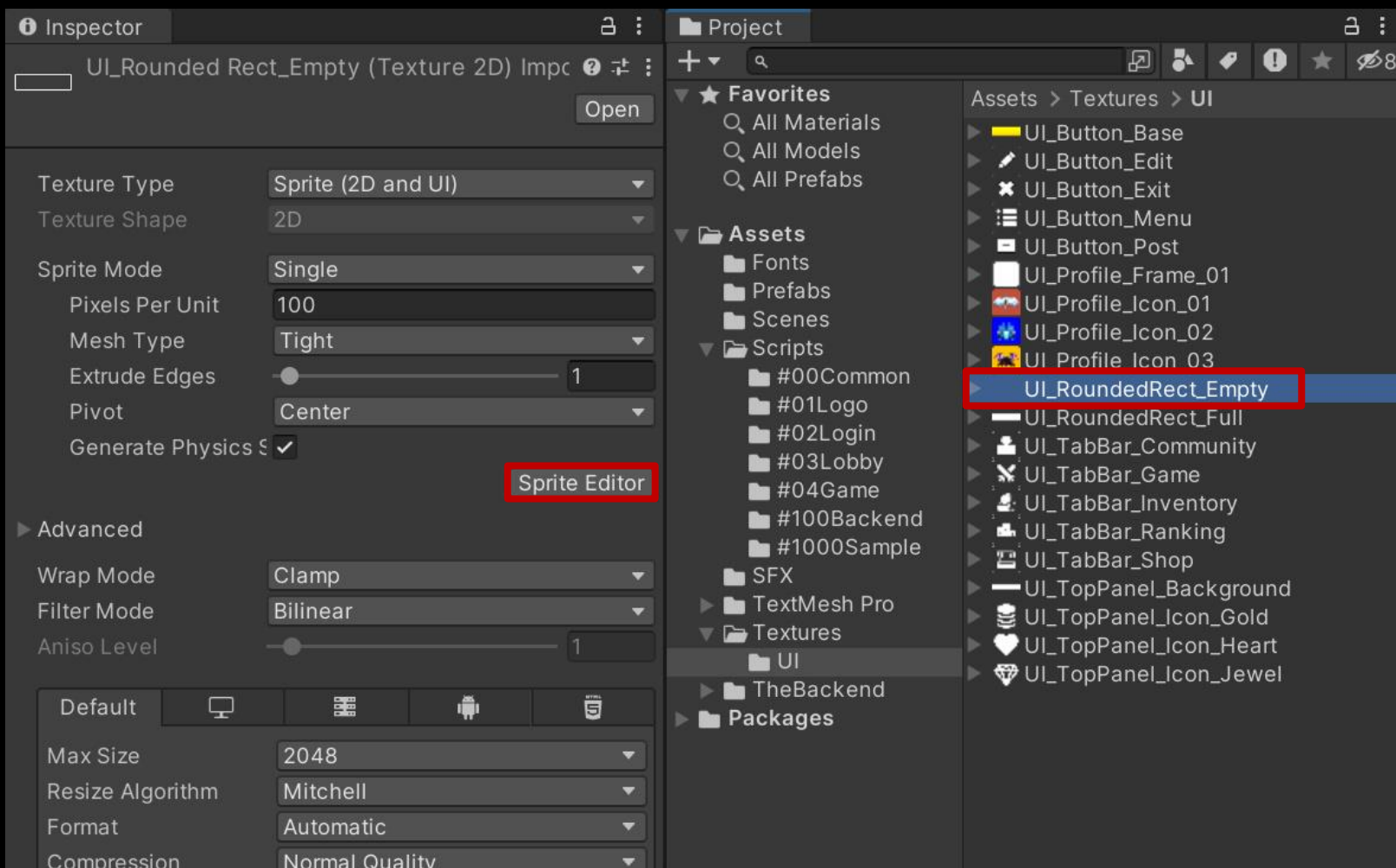
- 우편 UI 오브젝트
- 우편 리스트 정보를 바탕으로 우편 UI 생성



# 우편 (Post)

## ■ 우편 UI 오브젝트

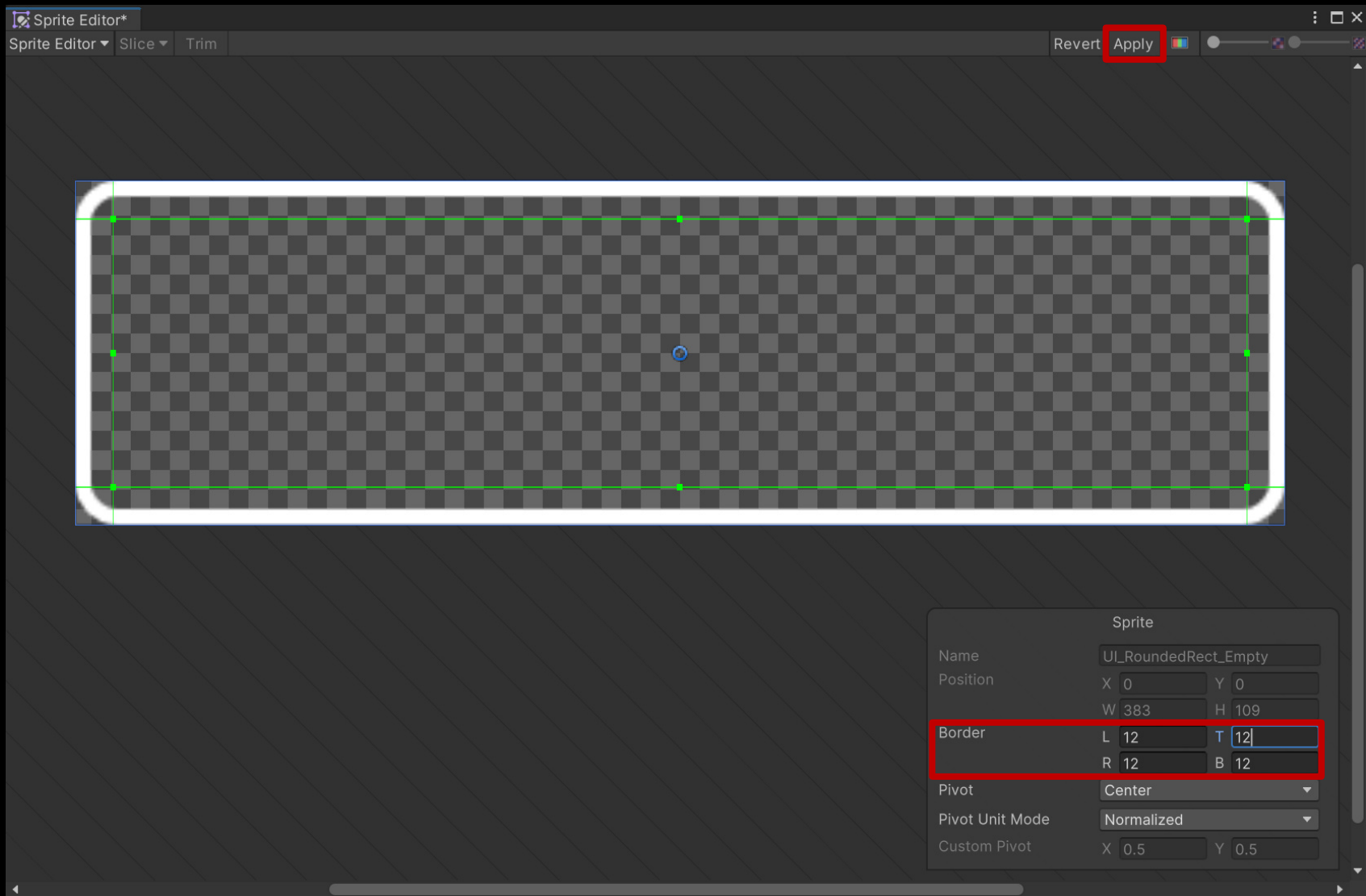
- UI\_RoundedRect\_Empty 이미지 에셋의 Border 설정





# 우편 (Post)

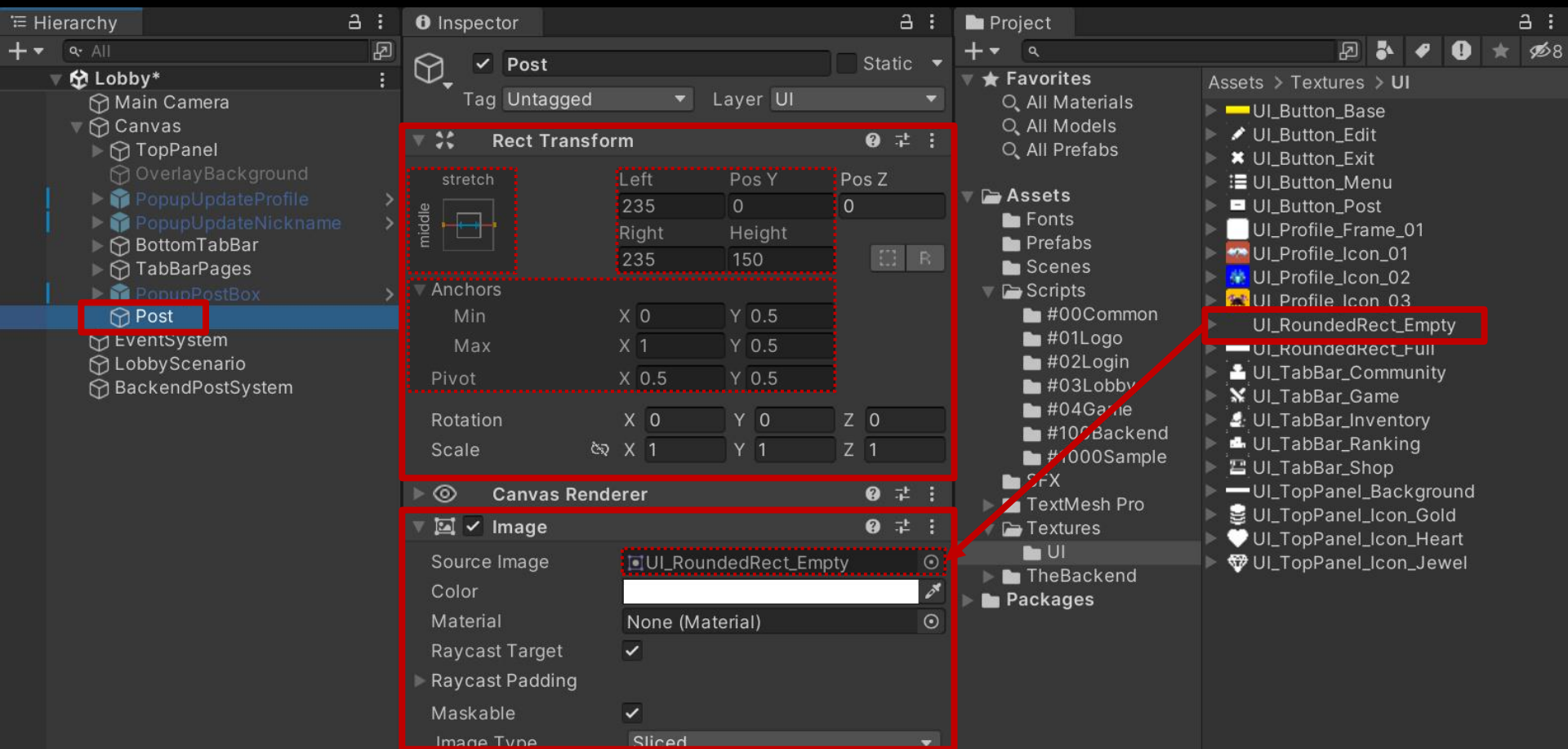
- UI\_RoundedRect\_Empty 이미지 에셋의 Border 설정 (계속)





# 우편 (Post)

- 우편 UI를 관리하는 Image UI 생성 및 설정
  - GameObject - UI - Image





# 우편 (Post)

- 아이템 정보의 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

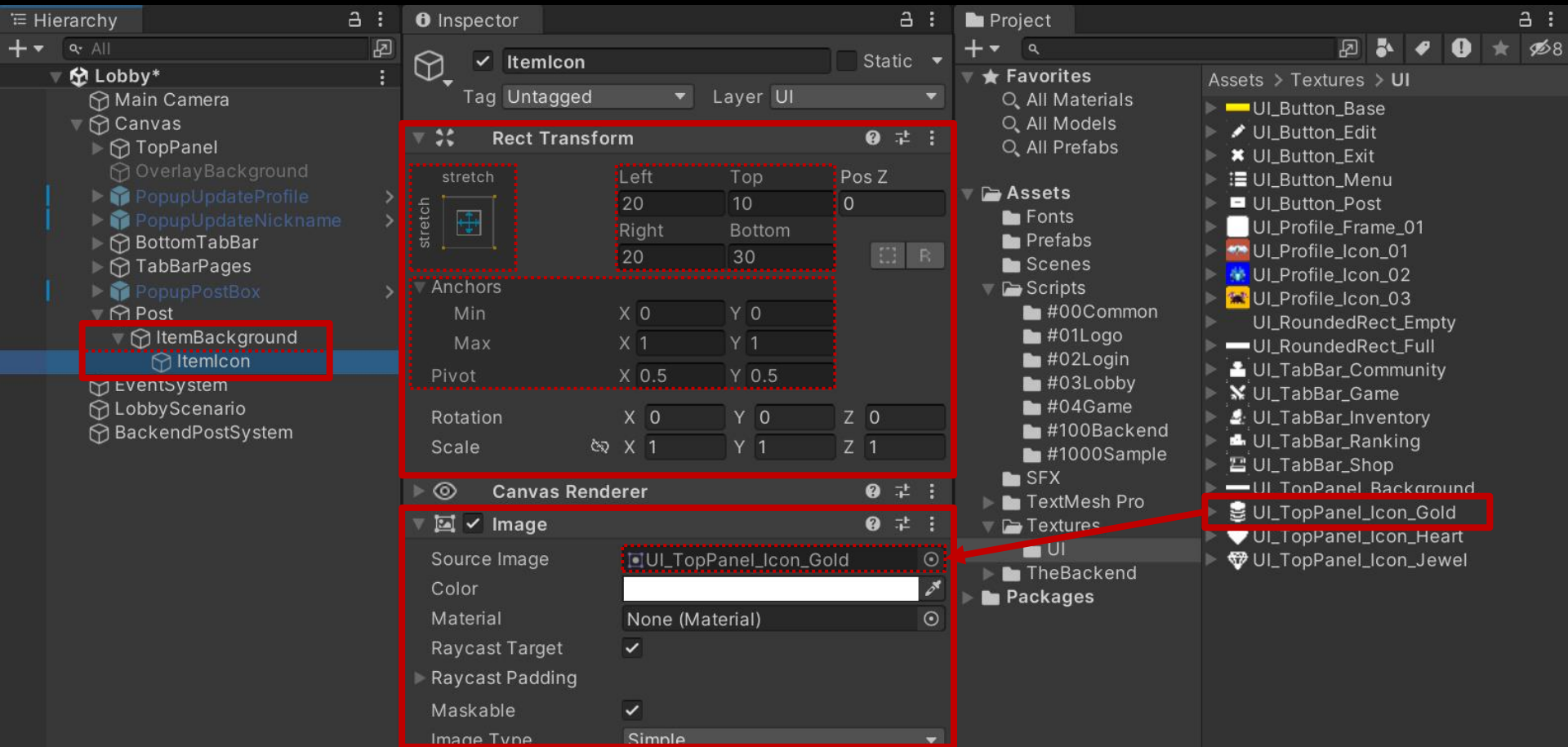
The screenshot displays the Unity Inspector for a GameObject named 'ItemBackground'. The Hierarchy panel on the left shows the object's path: Lobby\* > Canvas > Post > ItemBackground. The Inspector panel is divided into several sections:

- ItemBackground**: Includes a 'Static' checkbox (unchecked), 'Tag' set to 'Untagged', and 'Layer' set to 'UI'.
- Rect Transform**: Shows a visual representation of the rectangle with 'left' and 'middle' anchors. The 'Pos X' is 5, 'Pos Y' is 0, and 'Pos Z' is 0. The 'Width' is 140 and 'Height' is 140. The 'Anchors' section shows 'Min' (X: 0, Y: 0.5), 'Max' (X: 0, Y: 0.5), and 'Pivot' (X: 0, Y: 0.5). The 'Rotation' is (X: 0, Y: 0, Z: 0) and 'Scale' is (X: 1, Y: 1, Z: 1).
- Canvas Renderer**: A standard component for rendering UI elements.
- Image**: Shows 'Source Image' set to 'None (Sprite)'. The 'Color' is highlighted with a red dashed box and is currently white. The 'Material' is 'None (Material)' with a 'Color (50, 50, 50, 255)' value. 'Raycast Target' is checked.



# 우편 (Post)

- 아이템 아이콘을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image





# 우편 (Post)

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

Hierarchy: Lobby\* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > PopupPostBox > Post > ItemBackground > ItemIcon > ItemCount

Inspector: ItemCount (Static) Tag: Untagged Layer: UI

Rect Transform: stretch: bottom, Left: 0, Pos Y: 0, Pos Z: 0, Right: 0, Height: 40

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

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

Text Input: 1000

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: 30

Auto Size:

Vertex Color:

Color Gradient:

Override Tags:

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

Alignment: Left

Wrapping: Enabled

Overflow: Overflow

Horizontal Mapping: Character

Vertical Mapping: Character





# 우편 (Post)

- 우편 제목을 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 'Post' object is selected, and its 'Title' child is highlighted with a red box. The Inspector panel shows the 'Title' component with the following properties:

Rect Transform			
stretch	Left	Pos Y	Pos Z
top	160	-5	0
	Right	Height	
	220	50	
Anchors			
Min	X 0	Y 1	
Max	X 1	Y 1	
Pivot	X 0.5	Y 1	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

The Inspector panel also shows the 'TextMeshPro - Text (UI)' component with the following properties:

- 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
- Vertex Color: Color (73, 105, 139, 255)

The screenshot shows the Unity Inspector panel for the 'TextMeshPro - Text (UI)' component. The text content is '게임 오픈 기념 보상!'. The Inspector panel shows the following properties:

- Text Input: 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 (73, 105, 139, 255)
- Color Gradient:
- Override Tags:
- Spacing Options (em):
  - Character: 0
  - Word: 0
  - Line: 0
  - Paragraph: 0
- Alignment:  (Left)
- Wrapping: Enabled
- Overflow: Overflow
- Horizontal Mapping: Character
- Vertical Mapping: Character





# 우편 (Post)

- 우편 내용을 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Text - TextMeshPro"

Hierarchy: Lobby\* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > PopupPostBox > Post > ItemBackground > Title > Content

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

Rect Transform:

stretch	Left	Pos Y	Pos Z
bottom	160	5	0
	Right	Height	
	220	80	

Anchors:

Min	X	Y
	0	0
Max	X	Y
	1	0

Pivot: X 0.5, Y 0

Rotation: X 0, Y 0, Z 0

Scale: X 1, Y 1, Z 1

Canvas Renderer

TextMeshPro - Text (UI)

NotoSansKR-Regular SDF Material (Material)

Shader: TextMeshPro/Distance Field

Add Component

Text Input: Enable RTL Editor

안녕하세요. GM 고박사입니다.  
게임을 다운로드해주셔서 감사합니다.

Text Style: Normal

Main Settings

Font Asset: NotoSansKR-Regular SDF (TMF)

Material Preset: NotoSansKR-Regular SDF Material

Font Style: B I U S ab AB SC

Font Size: 30

Auto Size:

Vertex Color:

Color Gradient:

Override Tags:

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

Alignment:  Left,  Center,  Right

Wrapping: Enabled

Overflow: Overflow

Horizontal Mapping: Character

Vertical Mapping: Character



# 우편 (Post)

- 우편 수령 "Button - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Button - TextMeshPro"

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the 'ButtonReceive' object under the 'Post' folder highlighted. The Inspector panel on the right shows the properties of the selected 'ButtonReceive' object.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - Post
        - ItemBackground
        - Title
        - Content
        - ButtonReceive (Selected)
        - Text (TMP)
- EventSystem
- LobbyScenario
- BackendPostSystem

**Inspector Panel:**

- ButtonReceive** (Static)
  - Tag: Untagged
  - Layer: UI
- Rect Transform**
  - Position: Pos X: -5, Pos Y: 0, Pos Z: 0
  - Size: Width: 200, Height: 140
  - Anchor: right, middle
  - 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: UISprite
  - Color: Color (73, 105, 139, 255)
  - Material: None (Mat)
  - Raycast Target:



# 우편 (Post)

- 우편 수령 "Button - TextMeshPro" UI 생성 및 설정 (계속)

The image shows the Unity Inspector and Hierarchy panels. The Hierarchy panel on the left shows the scene structure, with 'ButtonReceive' and 'ReceiveText' highlighted. The Inspector panel in the center shows the 'ReceiveText' component selected, with the 'Rect Transform' component expanded. The 'Rect Transform' component has the following settings:

Property	Value
stretch	Left: 10, Pos Y: -10, Pos Z: 0
Right	Height: 80
Bottom	10
Anchor Min	X: 0, Y: 1
Anchor Max	X: 1, Y: 1
Pivot	X: 0.5, Y: 1
Rotation	X: 0, Y: 0, Z: 0
Scale	X: 1, Y: 1, Z: 1

The 'TextMeshPro - Text (UI)' component is also visible in the Inspector, with the following settings:

- Text Input: Enable RTL Editor (unchecked)
- 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: 30
  - Auto Size: (unchecked)
- Vertex Color: Color (255, 255, 255, 255)
- Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0
- Alignment: (Left-aligned)
- Wrapping: Enabled
- Overflow: Overflow
- Horizontal Mapping: Character
- Vertical Mapping: Character



# 우편 (Post)

- 우편 만료 날짜 배경을 출력하는 Image UI 생성 및 설정
  - GameObject - UI - Image

The screenshot displays the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the scene structure, with the 'DateBackground' object selected under the 'Post' folder. The Inspector panel on the right shows the properties of the selected object, which is a 'DateBackground' (a subclass of 'Image').

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - Post
        - ItemBackground
        - Title
        - Content
        - ButtonReceive
          - ReceiveText
          - DateBackground**

- EventSystem
- LobbyScenario
- BackendPostSystem

**Inspector Panel:**

- DateBackground** (Static)
  - Tag: Untagged
  - Layer: UI
- Rect Transform**
  - stretch: bottom (indicated by a red dashed box)
  - Left: 10
  - Pos Y: 10
  - Pos Z: 0
  - Right: 10
  - Height: 40
- Anchors**
  - Min: X 0, Y 0
  - Max: X 1, Y 0
  - Pivot: X 0.5, Y 0
- Rotation: X 0, Y 0, Z 0
- Scale: X 1, Y 1, Z 1

- Canvas Renderer**
- Image**
- Source Image: None (Sprite)
- Color: **Color (50, 50, 50, 255)** (indicated by a red dashed box)
- Material: None (Material)
- Raycast Target:



# 우편 (Post)

- 우편 만료 날짜를 출력하는 "Text - TextMeshPro" UI 생성 및 설정
  - GameObject - UI - "Text - TextMeshPro"

Hierarchy: Lobby\* > Canvas > TopPanel > OverlayBackground > PopupUpdateProfile > PopupUpdateNickname > BottomTabBar > TabBarPages > PopupPostBox > Post > ItemBackground > Title > Content > ButtonReceive > ReceiveText > DateBackground > ExpirationDate

Inspector: ExpirationDate (Static) | Tag: Untagged | Layer: UI

Rect Transform: stretch (Left, Top, Right, Bottom) | Pos Z: 0

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

Canvas Renderer

TextMeshPro - Text (UI)

NotoSansKR-Bold SDF Material (Material) | Shader: TextMeshPro/Distance Fi

Add Component

Text Input: 23시간 후 만료 | 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: 20 | Auto Size

Vertex Color: [Color Picker]

Color Gradient: [Color Gradient]

Override Tags: [Override Tags]

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

Alignment: [Alignment Icons]

Wrapping: Enabled

Overflow: Overflow

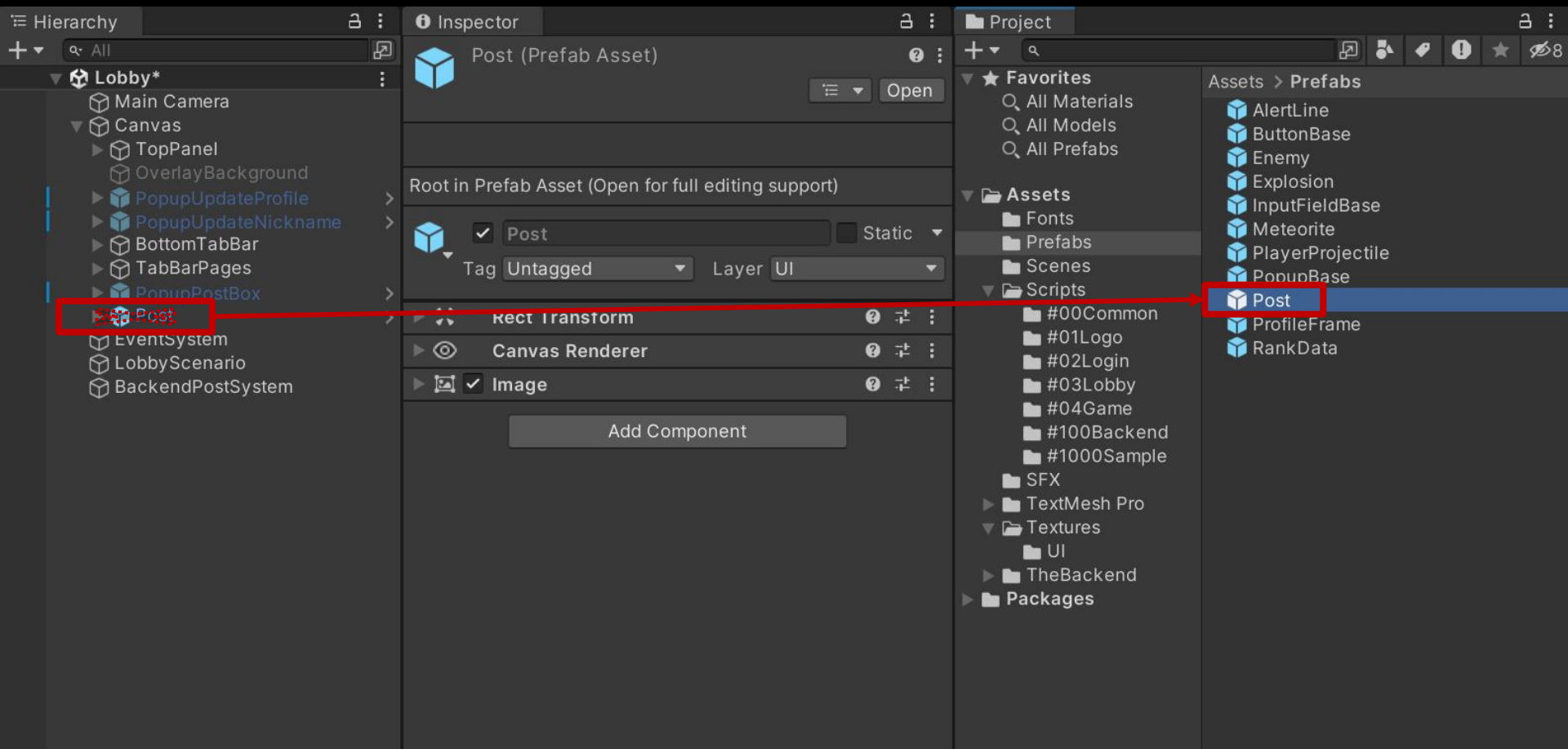
Horizontal Mapping: Character

Vertical Mapping: Character



# 우편 (Post)

- Post 오브젝트 Prefab 생성
  - Hierarchy View의 "Post" 오브젝트를 Project View로 Drag & Drop
  - Hierarchy View에 있는 "Post" 오브젝트 삭제







# 우편 (Post)

## ■ 우편 리스트 정보를 바탕으로 우편 UI 생성

- 우편 리스트 불러오기가 완료되었을 때 이벤트 메소드 호출

- BackendPostSystem Script 수정

```
1  using System.Collections.Generic;
2  using UnityEngine;
3  using UnityEngine.Events;
4  using BackEnd;
5
6  public class BackendPostSystem : MonoBehaviour
7  {
8      [System.Serializable]
9      public class PostEvent : UnityEvent<List<PostData>> { }
10     public PostEvent onGetPostListEvent = new PostEvent();
11
12     private List<PostData> postList = new List<PostData>();
13
14     /* private void Update() ... */
15
16     public void PostListGet()
17     {
18         PostListGet(PostType.Admin);
19     }
20
21     private void PostListGet(PostType type)
22     {
23         // ...
24     }
25
26     private void PostListUpdate()
27     {
28         // ...
29     }
30
31     private void PostListDelete()
32     {
33         // ...
34     }
35 }
```



# 우편 (Post)

## BackendPostSystem Script 수정 (계속)

```
35 public void PostListGet(PostType postType)
36 {
37     Backend.UPost.GetPostList(postType, callback =>
38     {
39         if ( !callback.IsSuccess() )...
44
45         Debug.Log($"우편 리스트 불러오기 요청에 성공했습니다 : {callback}");
46
47         // JSON 데이터 파싱 성공
48         try
49         {
50             LitJson.JsonData jsonData = callback.GetFlattenJSON()["postList"];
51
52             // 받은 데이터의 개수가 0이면 데이터가 없는 것
53             if ( jsonData.Count <= 0 )...
58
59             // 매번 우편 리스트를 불러올 때 postList 초기화
60             postList.Clear();
61
62             // 현재 저장 가능한 모든 우편 정보 불러오기
63             for ( int i = 0; i < jsonData.Count; ++ i )...
105
106             // 우편 리스트 불러오기가 완료되었을 때 이벤트 메소드 호출
107             onGetPostListEvent?.Invoke(postList);
108
109             // 저장 가능한 모든 우편(postList) 정보 출력
110             for ( int i = 0; i < postList.Count; ++ i )...
114         }
115         // JSON 데이터 파싱 실패
116         catch ( System.Exception e )
117         {
118             // try-catch 에러 출력
119             Debug.LogError(e);
120         }
121     });
122 }
```





# 우편 (Post)

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

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the **ButtonPost** object is selected and highlighted with a red box. Below it, the **BackendPostSystem** object is also highlighted with a red box. A red arrow points from the **BackendPostSystem** object to the **BackendPostSystem.PostListGet** event in the Inspector's **On Click ()** list.

The Inspector panel shows the **ButtonPost** object with the following properties:

- Tag:** Untagged
- Layer:** UI
- Rect Transform:** (Visible)
- Canvas Renderer:** (Visible)
- Image:** (Visible)
- Button:** (Visible)

The **Button** component properties are:

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

The **On Click ()** event list contains the following entries:

- Runtime Only | RectTransform.SetAsLastSibling | OverlayBa
- Runtime Only | GameObject.SetActive | OverlayBa
- Runtime Only | RectTransform.SetAsLastSibling | PopupPost
- Runtime Only | GameObject.SetActive | PopupPost
- Runtime Only | BackendPostSystem.PostListGet | BackendP



# 우편 (Post)

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

```
1 using System.Collections.Generic;
2 using UnityEngine;
3
4 public class PopupPostBox : MonoBehaviour
5 {
6     [SerializeField]
7     private GameObject      postPrefab;           // 우편 UI 프리팹
8     [SerializeField]
9     private Transform       parentContent;        // 우편 UI가 배치되는 ScrollView의 Content
10    [SerializeField]
11    private GameObject       textSystem;          // "우편함이 비어있습니다" 텍스트 오브젝트
12
13    private List<GameObject> postList;
14
15    private void Awake()
16    {
17        postList = new List<GameObject>();
18    }
19
20    private void OnDisable()
21    {
22        DestroyPostAll();
23    }
24
```



# 우편 (Post)

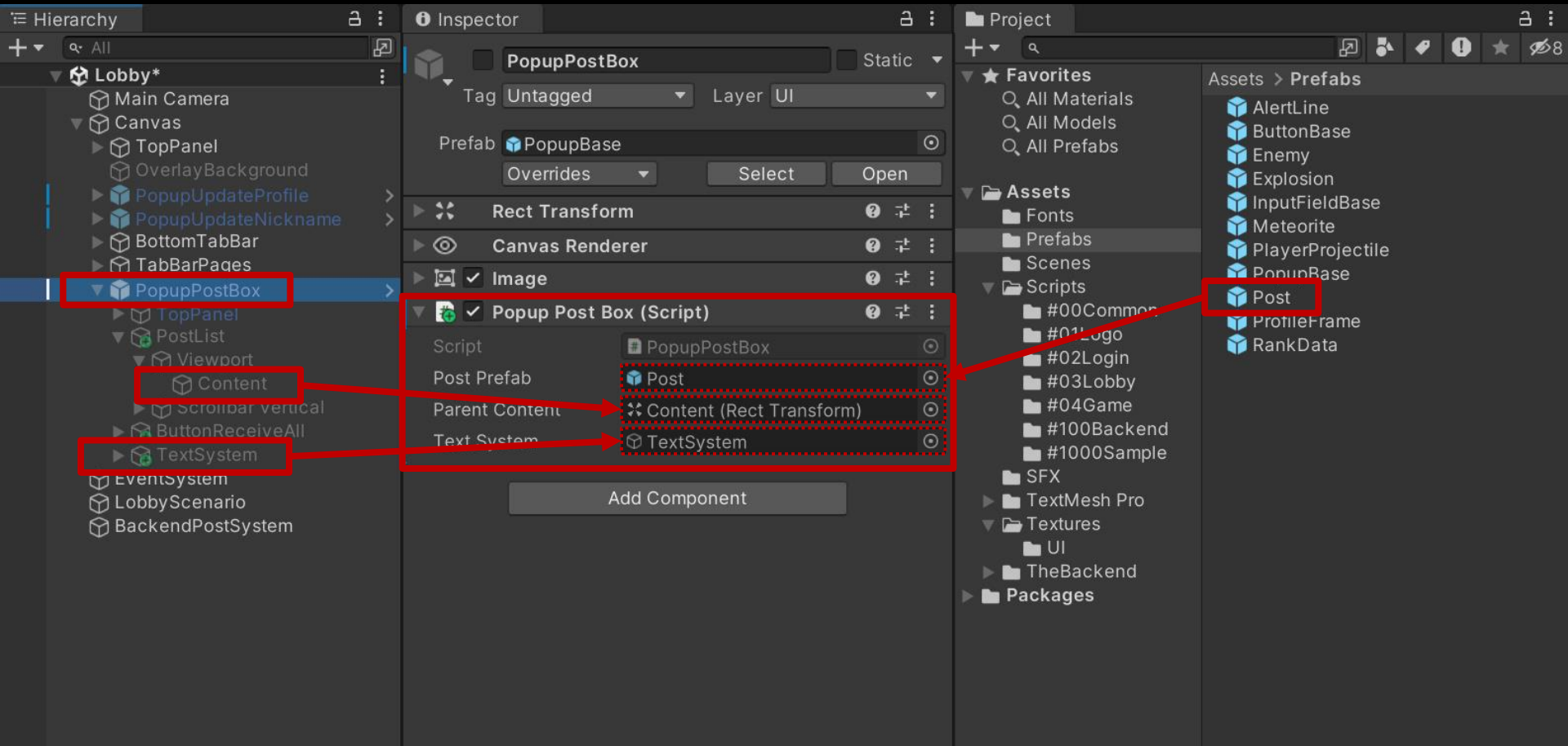
- 우편함 UI를 제어하는 스크립트 생성 및 작성 (계속)

```
25     public void SpawnPostAll(List<PostData> postDataList)
26     {
27         for ( int i = 0; i < postDataList.Count; ++ i )
28         {
29             GameObject clone = Instantiate(postPrefab, parentContent);
30
31             postList.Add(clone);
32         }
33
34         textSystem.SetActive(false);
35     }
36
37     public void DestroyPostAll()
38     {
39         foreach ( GameObject post in postList )
40         {
41             if ( post != null ) Destroy(post);
42         }
43
44         postList.Clear();
45
46         textSystem.SetActive(true);
47     }
48 }
```



# 우편 (Post)

- PopupPostBox 오브젝트에 "PopupPostBox" 컴포넌트 추가 및 설정





# 우편 (Post)

- BackendPostSystem 오브젝트의 "BackendPostSystem" 컴포넌트 변수 설정

The screenshot shows the Unity Inspector window with the following details:

- Hierarchy:** Shows a tree view of the scene objects. The **BackendPostSystem** component is highlighted in blue at the bottom. A red box highlights the **PopupPostBox** component in the hierarchy, with a red arrow pointing to the **Runtime Only** dropdown in the Inspector.
- Inspector:** Shows the **BackendPostSystem** component selected. The **Script** dropdown is set to **BackendPostSystem**. The **On Get Post List Event (List`1)** dropdown is set to **Runtime Only**. The **PopupPostBox.SpawnPostAll** dropdown is set to **PopupPos**. A red box highlights the **Runtime Only** dropdown, and another red box highlights the **PopupPos** dropdown.
- Project:** Shows the Project window with the **Assets** folder expanded, showing the **Scripts** folder and its contents.

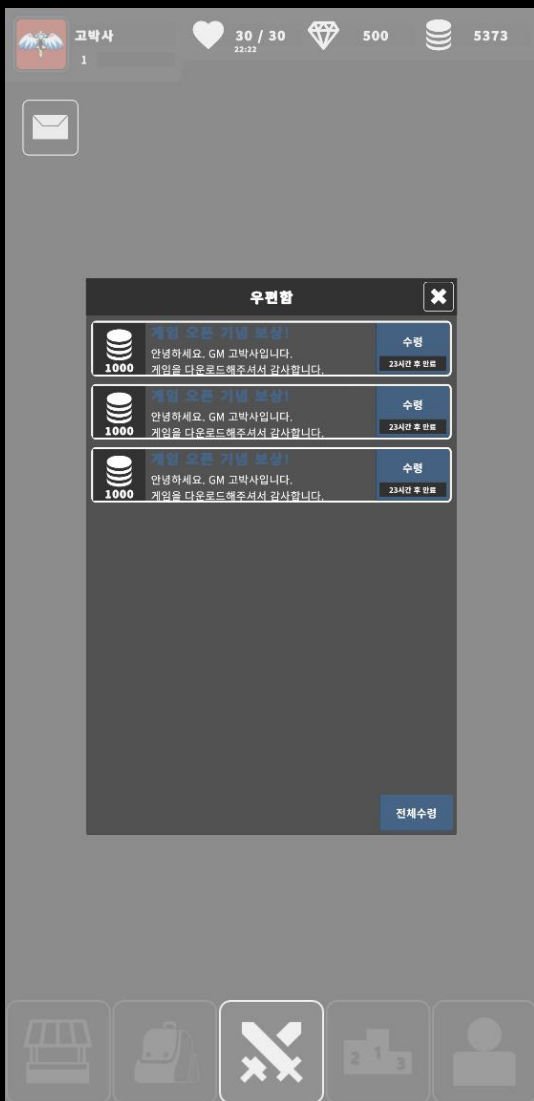
The **PopupPostBox** dropdown menu is open, showing the following options:

- No Function
- GameObject >
- RectTransform >
- CanvasRenderer >
- Image >
- PopupPostBox >
- Dynamic List`1
- SpawnPostAll
- Static Parameters
- bool enabled
- string name
- bool runInEditMode
- string tag
- bool useGUILayout
- BroadcastMessage (string)
- CancelInvoke (string)
- CancelInvoke ()
- DestroyPostAll ()
- SendMessage (string)
- SendMessageUpwards (string)
- StopAllCoroutines ()

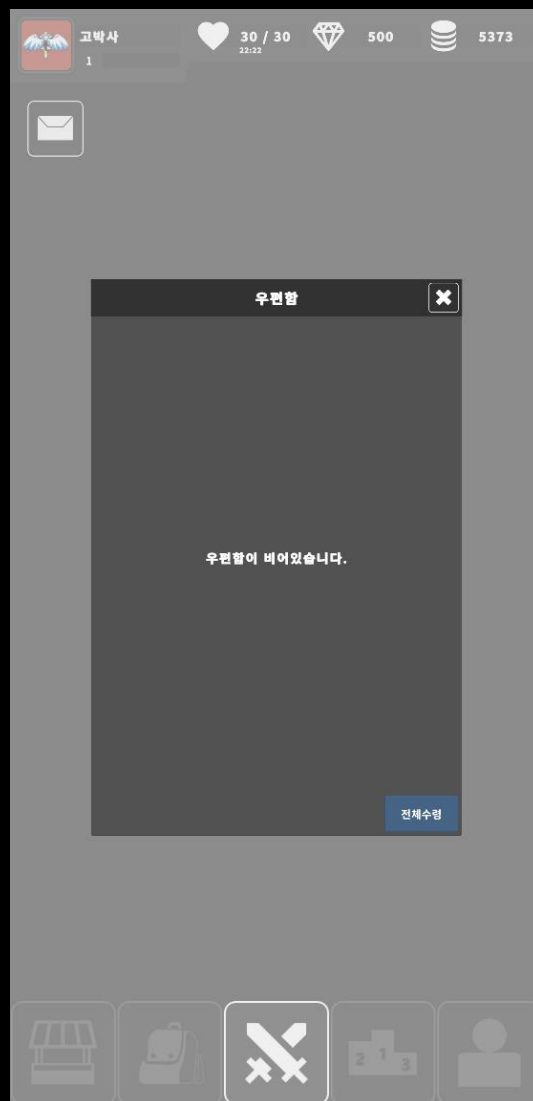


# 우편 (Post)

## ■ 결과 화면



< 우편이 있을 때 >

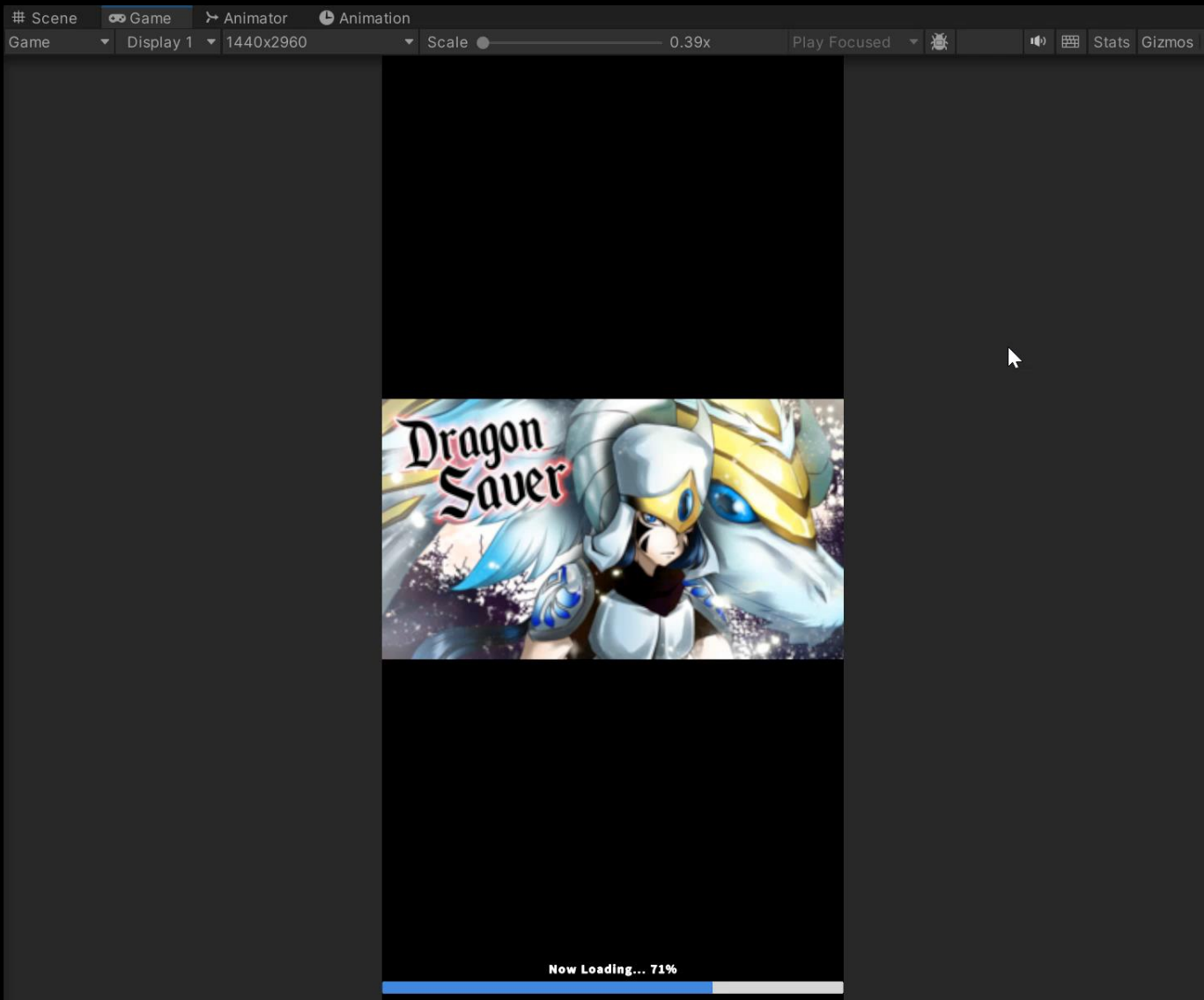


< 우편이 없을 때 >



# 우편 (Post)

## ■ 결과 화면



# 우편 데이터와 UI 연동

- 우편 데이터와 UI 연동
- 우편 하나 수령
- 우편 전체 수령





# 우편 데이터와 UI 연동

## ■ 우편 데이터와 UI 연동

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

```
1 using UnityEngine;
2 using UnityEngine.UI;
3 using TMPro;
4 using BackEnd;
5 using System;          // TimeSpan
6
7 public class Post : MonoBehaviour
8 {
9     [SerializeField]
10    private Sprite[]    spriteItemIcons;    // 우편에 포함된 아이템 아이콘에 출력할 이미지 배열
11    [SerializeField]
12    private Image       imageItemIcon;     // 우편에 포함된 아이템 아이콘 출력
13    [SerializeField]
14    private TextMeshProUGUI textItemCount; // 우편에 포함된 아이템의 개수
15    [SerializeField]
16    private TextMeshProUGUI textTitle;    // 우편 제목
17    [SerializeField]
18    private TextMeshProUGUI textContent;  // 우편 내용
19    [SerializeField]
20    private TextMeshProUGUI textExpirationDate; // 우편 만료까지 남은 시간 출력
21
```



# 우편 데이터와 UI 연동

## ■ 우편 UI를 제어하는 스크립트 생성 및 작성 (계속)

```
22 public void Setup(PostData postData)
23 {
24     // 우편 제목과 내용 설정
25     textTitle.text = postData.title;
26     textContent.text = postData.content;
27
28     // 첫 번째 아이템 정보를 우편에 출력
29     foreach ( string itemKey in postData.postReward.Keys )
30     {
31         // 우편에 포함된 아이템 이미지 출력
32         if ( itemKey.Equals("heart") )      imageItemIcon.sprite = spriteItemIcons[0];
33         else if ( itemKey.Equals("gold") )  imageItemIcon.sprite = spriteItemIcons[1];
34         else if ( itemKey.Equals("jewel") ) imageItemIcon.sprite = spriteItemIcons[2];
35
36         // 우편에 포함된 아이템 개수 출력
37         textItemCount.text = postData.postReward[itemKey].ToString();
38
39         // 하나의 우편에 포함된 아이템이 여러 개 일 수도 있는데 현재 예제에서는 첫 번째 아이템 정보만 출력
40         break;
41     }
42 }
```



# 우편 데이터와 UI 연동

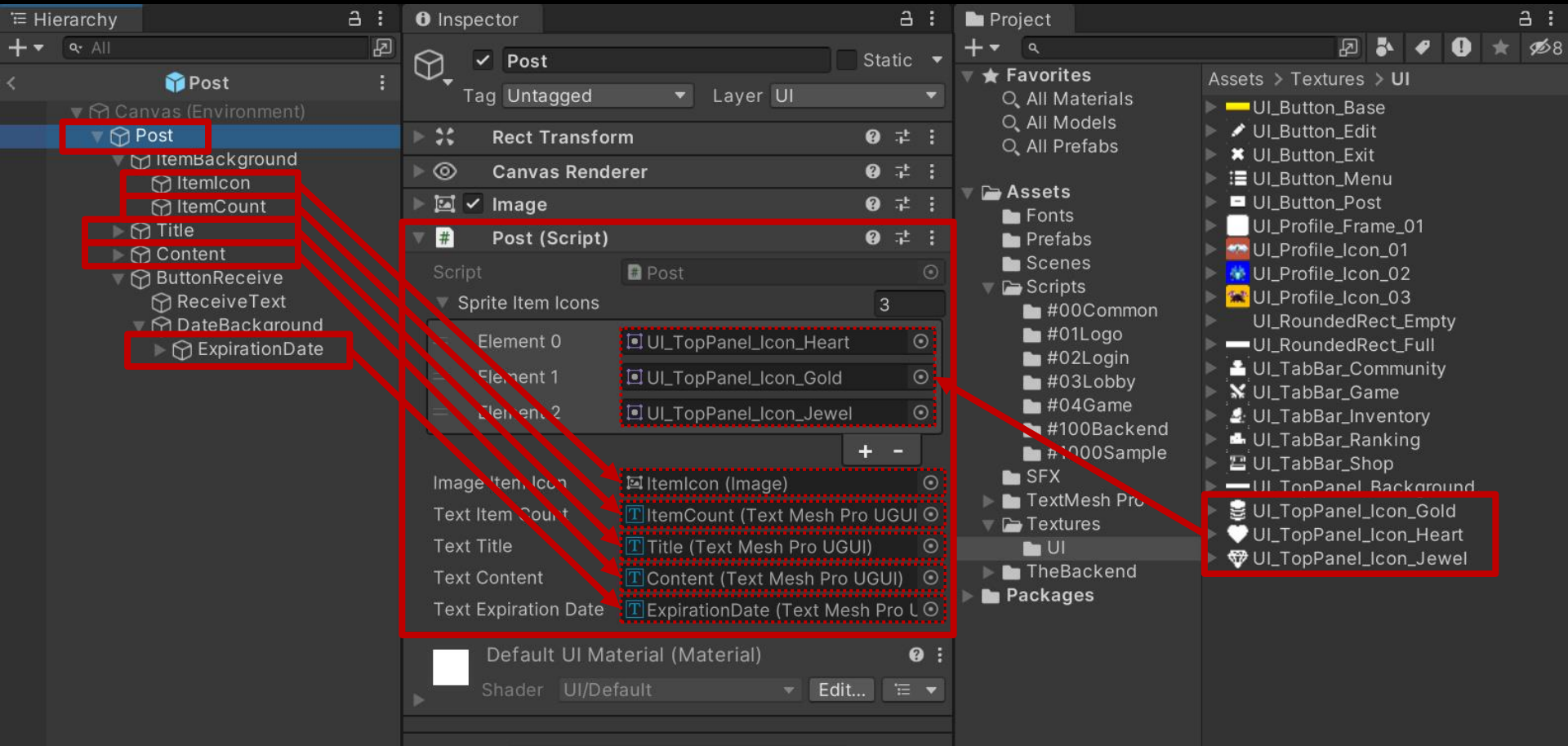
## ■ 우편 UI를 제어하는 스크립트 생성 및 작성 (계속)

```
43 // GetServerTime() - 서버 시간 불러오기
44 Backend.Utils.GetServerTime(callback =>
45 {
46     if ( !callback.IsSuccess() )
47     {
48         Debug.LogError($"서버 시간 불러오기에 실패했습니다. : {callback}");
49         return;
50     }
51
52     // JSON 데이터 파싱 성공
53     try
54     {
55         // 현재 서버 시간
56         string serverTime = callback.GetFlattenJSON()["utcTime"].ToString();
57
58         // 우편 만료까지 남은 시간 = 우편 만료 시간 - 현재 서버 시간
59         TimeSpan timeSpan = DateTime.Parse(postData.expirationDate) - DateTime.Parse(serverTime);
60
61         // timeSpan.TotalHours로 남은 기간을 시(hour)로 표현
62         textExpirationDate.text = $"{timeSpan.TotalHours:F0}시간 후 만료";
63     }
64     // JSON 데이터 파싱 실패
65     catch ( Exception e )
66     {
67         // try-catch 에러 출력
68         Debug.LogError(e);
69     }
70 });
71 }
72 }
```



# 우편 데이터와 UI 연동

- Post 프리팹에 "Post" 컴포넌트 추가 및 설정







# 우편 데이터와 UI 연동

- 우편 UI를 생성할 때 Setup() 메소드를 호출해 우편 데이터 설정
  - PopupPostBox Script 수정

```
1  using ...
3
4  public class PopupPostBox : MonoBehaviour
5  {
6      [SerializeField]
7      private GameObject      postPrefab;      // 우편 UI 프리팹
8      [SerializeField]
9      private Transform      parentContent;    // 우편 UI가 배치되는 ScrollView의 Content
10     [SerializeField]
11     private GameObject      textSystem;      // "우편함이 비어있습니다" 텍스트 오브젝트
12
13     private List<GameObject> postList;
14
15     private void Awake()...
19
20     private void OnDisable()...
24
25     public void SpawnPostAll(List<PostData> postDataList)
26     {
27         for ( int i = 0; i < postDataList.Count; ++ i )
28         {
29             GameObject clone = Instantiate(postPrefab, parentContent);
30             clone.GetComponent<Post>().Setup(postDataList[i]);
31
32             postList.Add(clone);
33         }
34
35         textSystem.SetActive(false);
36     }
37
38     public void DestroyPostAll()...
49 }
```



# 우편 데이터와 UI 연동

## ■ 결과 화면

The screenshot displays the Unity 2022.2.2f1 interface for a game titled 'BackendBase - Lobby'. The main view shows a dark-themed lobby with a yellow '게임시작' (Start Game) button in the center. The top UI bar includes a character icon, a heart icon with '36 / 30', a diamond icon with '500', and a stack of coins icon with '5373'. Below the main view, there are five icons representing different game elements: a bus, a backpack, a sword, a building, and a person. The console window at the bottom left shows the following log messages:

```
[11:27:00] 2번째 우편  
title : 2  
[11:27:09] 우편 리스트 불러오기 요청에 성공했습니다 : statusCode : 200  
message : Success  
[11:27:09] 0번째 우편  
title : 오만 기쁨!  
[11:27:09] 1번째 우편  
title : 2  
[11:27:09] 2번째 우편  
title : 1
```

The right side of the interface shows the Hierarchy panel with 'Lobby' and 'DontDestroyOnLoad', the Inspector panel showing 'Logo (Scene Asset)', and the Project panel with a tree view of assets including Scripts, SFX, TextMesh Pro, Textures, UI, and Packages.



# 우편 데이터와 UI 연동

## ■ 우편 하나 수령

- inDate 정보를 매개변수로 받아 우편을 수령하는 PostReceive() 메소드 호출

- BackendPostSystem Script 수정

```
1  +using ...
5
6  -public class BackendPostSystem : MonoBehaviour
7  {
8      [System.Serializable]
9      public class PostEvent : UnityEvent<PostData>
10     public PostEvent onGetPostListEvent;
11
12     private List<PostData> postList;
13
14     +public void PostListGet()...
18
19     -public void PostReceive(PostType postType, string inDate)
20     {
21         PostReceive(postType, postList.FindIndex(item => item.inDate.Equals(inDate)));
22     }
23
24     +public void PostListGet(PostType postType)...
112
113    +public void PostReceive(PostType postType, int index)...
155
156    +public void PostReceiveAll(PostType postType)...
188
189    +public void SavePostToLocal(LitJson.JsonData item)...
227 }
```

```
public void PostReceive(PostType postType, string inDate)
{
    for ( int i = 0; i < postList.Count; ++ i )
    {
        if ( postList[i].inDate.Equals(inDate) )
        {
            PostReceive(postType, i);
            return;
        }
    }
}
```

동일한 결과의 코드입니다.





# 우편 데이터와 UI 연동

- 우편 UI를 삭제하는 DestroyPost() 메소드 정의
  - PopupPostBox Script 수정

```
1  +using ...
3
4  public class PopupPostBox : MonoBehaviour
5  {
6      [SerializeField]
7      private BackendPostSystem backendPostSystem; // 우편 "수령" 버튼을 눌렀을 때 PostReceive() 호출용
8      [SerializeField]
9      private GameObject postPrefab; // 우편 UI 프리팹
10     [SerializeField]
11     private Transform parentContent; // 우편 UI가 배치되는 ScrollView의 Content
12     [SerializeField]
13     private GameObject textSystem; // "우편함이 비어있습니다" 텍스트 오브젝트
14
15     private List<GameObject> postList;
16
17     + private void Awake()...
21
22     + private void OnDisable()...
26
```



# 우편 데이터와 UI 연동

## □ PopupPostBox Script 수정 (계속)

```
27 public void SpawnPostAll(List<PostData> postDataList)
28 {
29     for ( int i = 0; i < postDataList.Count; ++ i )
30     {
31         GameObject clone = Instantiate(postPrefab, parentContent);
32         clone.GetComponent<Post>().Setup(backendPostSystem, this, postDataList[i]);
33     }
34     postList.Add(clone);
35 }
36
37 textSystem.SetActive(false);
38 }
39
40 public void DestroyPostAll()...
41
42 public void DestroyPost(GameObject post)
43 {
44     Destroy(post);
45     postList.Remove(post);
46
47     if ( postList.Count == 0 )
48     {
49         textSystem.SetActive(true);
50     }
51 }
52
53
54
55
56
57
58
59
60
61
62 }
```



# 우편 데이터와 UI 연동

- 우편 수령 버튼을 눌렀을 때 호출하는 `OnClickPostReceive()` 메소드 정의
  - Post Script 수정

```
1  using ... // TimeSpan
2
3
4
5
6
7  public class Post : MonoBehaviour
8  {
9      [SerializeField]
10     private Sprite[]      spriteItemIcons; // 우편에 포함된 아이템 아이콘에 출력할 이미지 배열
11     [SerializeField]
12     private Image        imageItemIcon; // 우편에 포함된 아이템 아이콘 출력
13     [SerializeField]
14     private TextMeshProUGUI textItemCount; // 우편에 포함된 아이템의 개수
15     [SerializeField]
16     private TextMeshProUGUI textTitle; // 우편 제목
17     [SerializeField]
18     private TextMeshProUGUI textContent; // 우편 내용
19     [SerializeField]
20     private TextMeshProUGUI textExpirationDate; // 우편 만료까지 남은 시간 출력
21
22     [SerializeField]
23     private Button        buttonReceive; // 우편 "수령" 버튼 처리
24
25     private BackendPostSystem backendPostSystem;
26     private PopupPostBox popupPostBox;
27     private PostData postData;
28 }
```



# 우편 데이터와 UI 연동

## □ Post Script 수정 (계속)

```
29 public void Setup(BackendPostSystem postSystem, PopupPostBox postBox, PostData postData)
30 {
31     // 우편 "수령" 버튼을 눌렀을 때 처리
32     buttonReceive.onClick.AddListener(OnClickPostReceive);
33
34     backendPostSystem = postSystem;
35     popupPostBox      = postBox;
36     this.postData     = postData;
37
38     // 우편 제목과 내용 설정
39     textTitle.text    = postData.title;
40     textContent.text  = postData.content;
41
42     // 첫 번째 아이템 정보를 우편에 출력
43     foreach ( string itemKey in postData.postReward.Keys ) {...}
44
45     // GetServerTime() - 서버 시간 불러오기
46     Backend.Utills.GetServerTime(callback => {...});
47
48 }
49
50 private void OnClickPostReceive()
51 {
52     // 현재 우편 UI 오브젝트 삭제
53     popupPostBox.DestroyPost(gameObject);
54     // 우편 수령
55     backendPostSystem.PostReceive(PostType.Admin, postData.inDate);
56 }
57 }
```



# 우편 데이터와 UI 연동

- PopupPostBox 오브젝트의 "PopupPostBox" 컴포넌트 변수 설정

The screenshot shows the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'Lobby\*' object is expanded, and the 'PopupPostBox' component is selected. In the Inspector panel, the 'PopupPostBox' component is selected, and its 'Backend Post System' property is set to 'BackendPostSystem (Backend F...)'.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox**
    - EventSystem
    - LobbyScenario
    - BackendPostSystem

**Inspector Panel:**

- Object: PopupPostBox (Static)
- Tag: Untagged, Layer: UI
- Prefab: PopupBase
- Overrides: [Dropdown], Select, Open
- Rect Transform
- Canvas Renderer
- Image
- Popup Post Box (Script)**
  - Script: PopupPostBox
  - Backend Post System: **BackendPostSystem (Backend F...)**
  - Post Prefab: Post
  - Parent Content: Content (Rect Transform)
  - Text System: TextSystem

**Add Component**





# 우편 데이터와 UI 연동

- Post 프리팹의 "Post" 컴포넌트 변수 설정

The screenshot displays the Unity Inspector for a 'Post' component. The 'Post (Script)' section is expanded, showing the following fields:

- Script: Post
- Sprite Item Icons: 3
- Image Item Icon: ItemIcon (Image)
- Text Item Count: ItemCount (Text Mesh Pro UGUI)
- Text Title: Title (Text Mesh Pro UGUI)
- Text Content: Content (Text Mesh Pro UGUI)
- Text Expiration Date: ExpirationDate (Text Mesh Pro UGUI)
- Button Receive: ButtonReceive (Button)

The 'Button Receive' field is highlighted with a red dashed box, and a red arrow points from the 'ButtonReceive' component in the Hierarchy panel to this field. The 'Assets > Prefabs' panel on the right shows a list of prefabs, including 'Post'.



# 우편 데이터와 UI 연동

## ■ 우편 전체 수령

- 모든 우편을 수령하는 PostReceiveAll() 메소드 호출

- BackendPostSystem Script 수정

```
1  using ...
5
6  public class BackendPostSystem : MonoBehaviour
7  {
8      [System.Serializable]
9      public class PostEvent : UnityEvent<List<PostData>> { }
10     public PostEvent onGetPostListEvent = new PostEvent();
11
12     private List<PostData> postList = new List<PostData>();
13
14     public void PostListGet()...
18
19     public void PostReceive(PostType postType, string inDate)...
23
24     public void PostReceiveAll()
25     {
26         PostReceiveAll(PostType.Admin);
27     }
28
29     public void PostListGet(PostType postType)...
117
118     public void PostReceive(PostType postType, int index)...
160
161     public void PostReceiveAll(PostType postType)...
193
194     public void SavePostToLocal(LitJson.JsonData item)...
232 }
```





# 우편 데이터와 UI 연동

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

The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows a tree structure under 'Lobby\*'. The 'ButtonReceiveAll' object is selected and highlighted in blue. Red boxes highlight the 'PopupPostBox', 'ButtonReceiveAll', and 'BackendPostSystem' objects in the hierarchy. The Inspector panel on the right shows the properties of the selected 'Button' component. The 'On Click ()' section is expanded, showing two event entries. The first entry is 'Runtime Only' with the method 'BackendPostSystem.PostReceive' and the target 'BackendPostSystem'. The second entry is 'Runtime Only' with the method 'PopupPostBox.DestroyPostAll' and the target 'PopupPostBox'. Red dashed boxes highlight these two event entries. A red arrow points from the 'ButtonReceiveAll' object in the hierarchy to the 'Button' component in the Inspector.

**Hierarchy Panel:**

- Lobby\*
  - Main Camera
  - Canvas
    - TopPanel
    - OverlayBackground
    - PopupUpdateProfile
    - PopupUpdateNickname
    - BottomTabBar
    - TabBarPages
    - PopupPostBox
      - TopPanel
      - PostList
      - ButtonReceiveAll**
      - TextSystem
    - EventSystem
    - LobbyScenario
    - BackendPostSystem

**Inspector Panel:**

- ButtonReceiveAll (Static)
  - Tag: Untagged
  - Layer: UI
  - Rect Transform
  - Canvas Renderer
  - Image
  - Button**
    - Interactable:
    - Transition: Color Tint
    - Target Graphic: ButtonReceiveAll (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
    - On Click ()
      - Runtime Only | BackendPostSystem.PostReceive | BackendPostSystem
      - Runtime Only | PopupPostBox.DestroyPostAll | PopupPostBox



# 우편 데이터와 UI 연동

## ■ 결과 화면

