

GUI 클래스

GUI 클래스

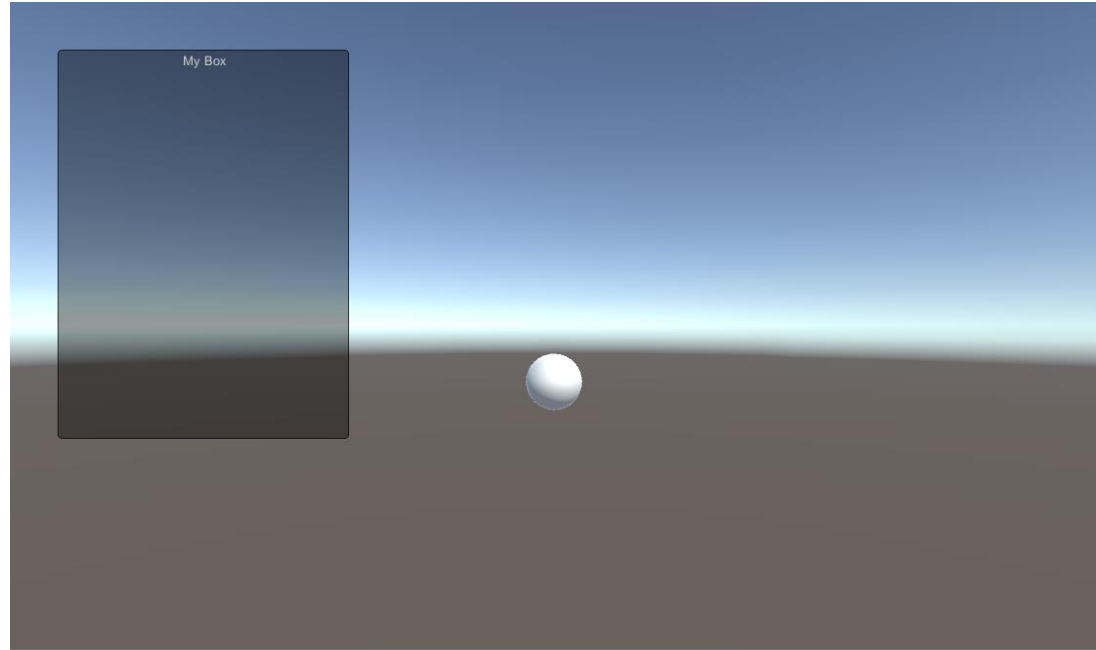
- GUI를 정의하고 있는 클래스
- 유니티에서 사용할 수 있는 GUI요소들을 스크립트에서 처리할 수 있다.
- 주요 멤버 Static 함수를 이용하여 씬에 GUI요소를 생성

```
GUI.컴포넌트명(위치, 내용,...);
```

박스

- 해당 영역에 박스를 표시

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUITest : MonoBehaviour {
5     void OnGUI () {
6         GUI.Box(new Rect(50, 50, 300, 200), "MyBox");
7     }
8 }
```



레이블

- 문자열이나 이미지 표시

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class UILabel : MonoBehaviour {
5
6     public Texture2D thumbnail;
7
8     void OnGUI() {
9         GUI.Label (new Rect (50, 50, 100, 30), "My Label");
10        GUI.Label (new Rect (50, 150, 100, 100), thumbnail);
11    }
12 }
```



버튼

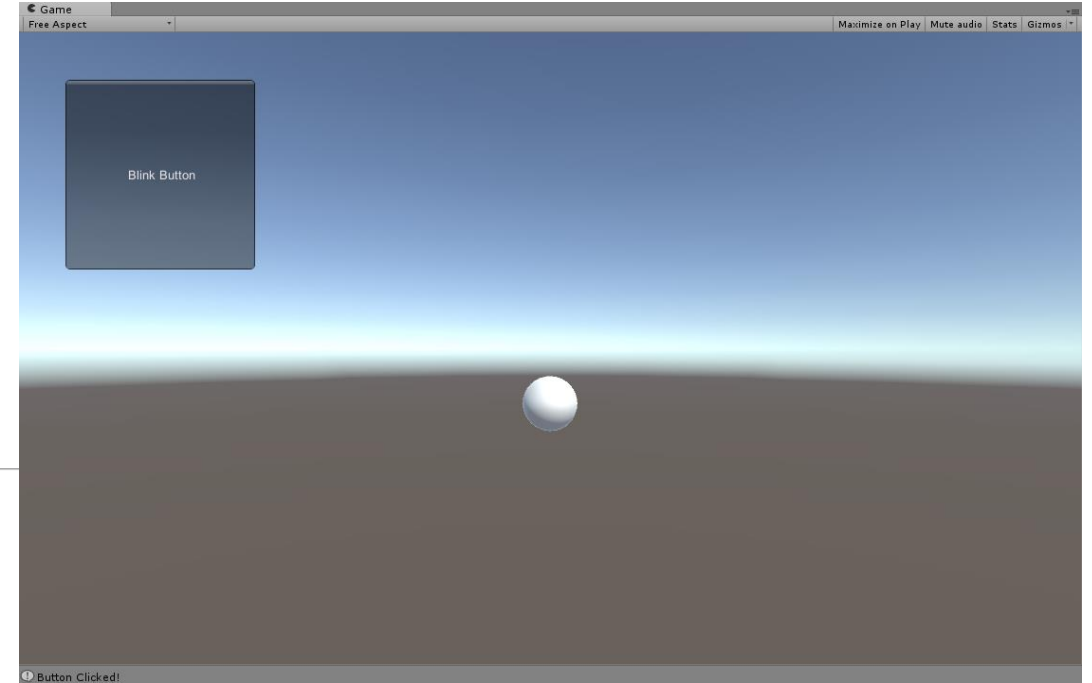
- 누를 때 마다 true 값을 반환하며 이벤트 발생

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIButton : MonoBehaviour {
5     void OnGUI() {
6         if (GUI.Button (new Rect (30, 40, 80, 20), "My Button")) {
7             Debug.Log("Button Clicked!");
8         }
9     }
10 }
```



버튼 깜박임

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIButtonBlink : MonoBehaviour {
5
6     void OnGUI() {
7         if (Time.time % 2 < 1) {
8             if (GUI.Button (new Rect (50, 50, 200, 200), "Blink Button")) {
9                 Debug.Log("Button Clicked!");
10            }
11        }
12    }
13 }
```



반복 버튼

- 누른채로 있으면 true 값을 발생하며 지속적으로 이벤트 발생

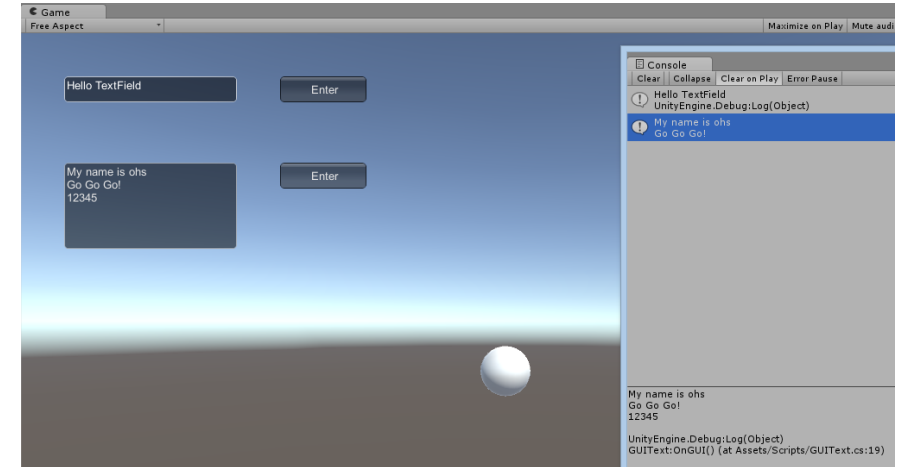
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIRepeatButton : MonoBehaviour {
5
6     void OnGUI() {
7         if (GUI.Button (new Rect (50, 50, 150, 30), "My Button")) {
8             Debug.Log(Time.time);
9         }
10
11         if (GUI.RepeatButton (new Rect (50, 150, 150, 30), "My RepeatButton")) {
12             Debug.Log(Time.time);
13         }
14     }
15 }
```



텍스트 필드와 텍스트 영역

- 텍스트 문자의 입력과 출력 처리

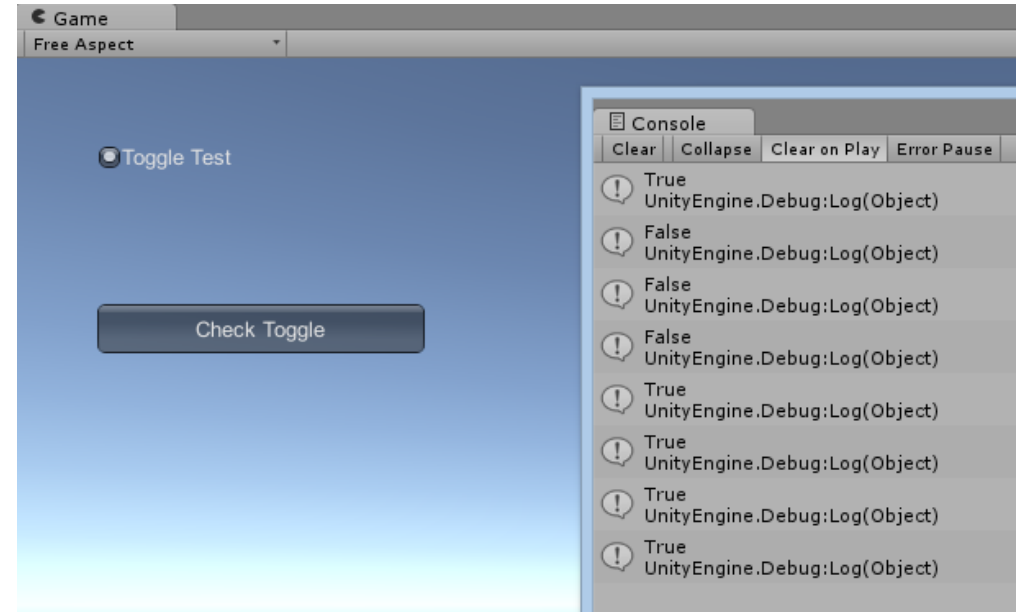
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIText : MonoBehaviour {
5
6     private string textFieldString = "input text";
7     private string textAreaString = "input text";
8
9     void OnGUI() {
10         textFieldString = GUI.TextField (new Rect (50, 50, 200, 30), textFieldString);
11
12         if (GUI.Button (new Rect (300, 50, 100, 30), "Enter")) {
13             Debug.Log(textFieldString);
14         }
15
16         textAreaString = GUI.TextArea (new Rect (50, 150, 200, 100), textAreaString);
17
18         if (GUI.Button (new Rect (300, 150, 100, 30), "Enter")) {
19             Debug.Log(textAreaString);
20         }
21     }
22 }
```



토글

- 특정 값의 On/Off 상태를 확인하는 체크박스

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIToggle : MonoBehaviour {
5
6     private bool toggle = true;
7
8     void OnGUI() {
9
10         toggle = GUI.Toggle (new Rect (50, 50, 200, 30), toggle, "Toggle Test");
11
12         if (GUI.Button (new Rect (50, 150, 200, 30), "Check Toggle")) {
13             Debug.Log(toggle);
14         }
15     }
16 }
```



툴바

- 횡으로 버튼들을 배열하여 그중 하나를 선택

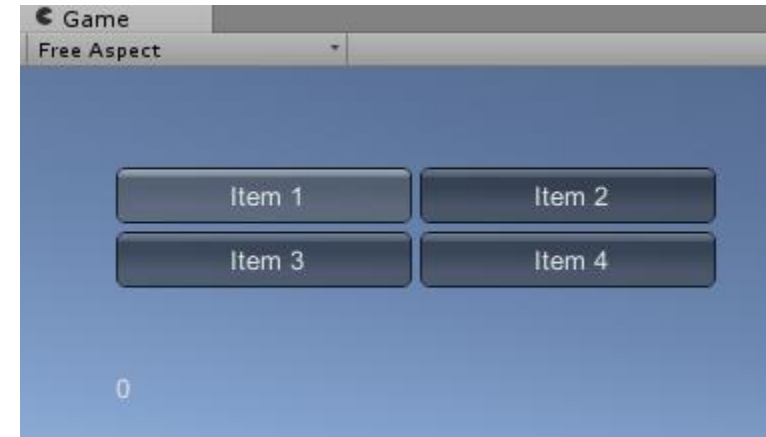
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIToolBar : MonoBehaviour {
5
6     private int toolbarIndex = 0;
7     private string[] toolbarItems = {"Item 1", "Item 2", "Item 3"};
8
9     void OnGUI() {
10         toolbarIndex = GUI.Toolbar (new Rect (50, 50, 300, 30), toolbarIndex, toolbarItems);
11         GUI.Label (new Rect (50, 150, 100, 30), toolbarIndex.ToString());
12     }
13 }
```



선택 그리드

- 격자형으로 버튼들을 배열하여 그중 하나를 선택

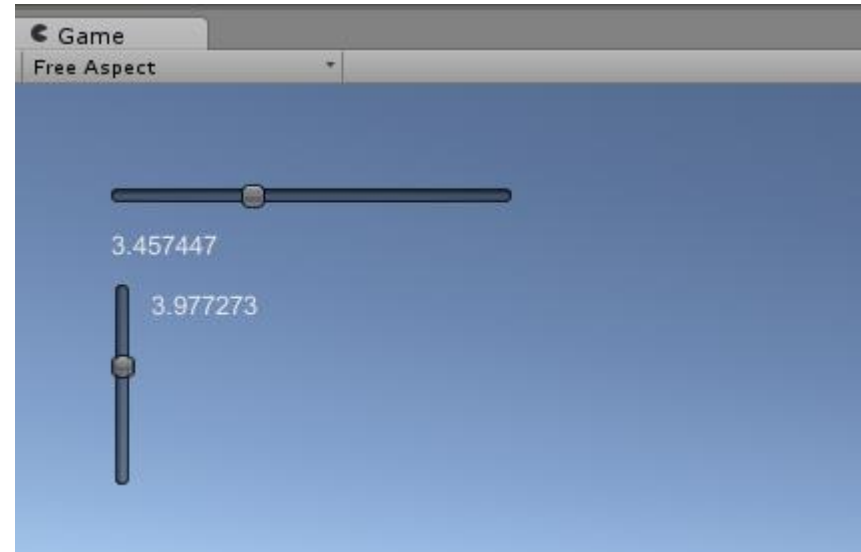
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUISelectionGrid : MonoBehaviour {
5
6     private int selectionGridIndex = 0;
7     private string[] selectionGridItems = {"Item 1", "Item 2", "Item 3", "Item 4"};
8
9     void OnGUI() {
10         selectionGridIndex = GUI.SelectionGrid (new Rect (50, 50, 300, 60), selectionGridIndex, selectionGridItems, 2);
11         GUI.Label (new Rect (50, 150, 100, 30), selectionGridIndex.ToString());
12     }
13 }
```



슬라이더

- 값을 조정하는데 이용하는 슬라이딩 바

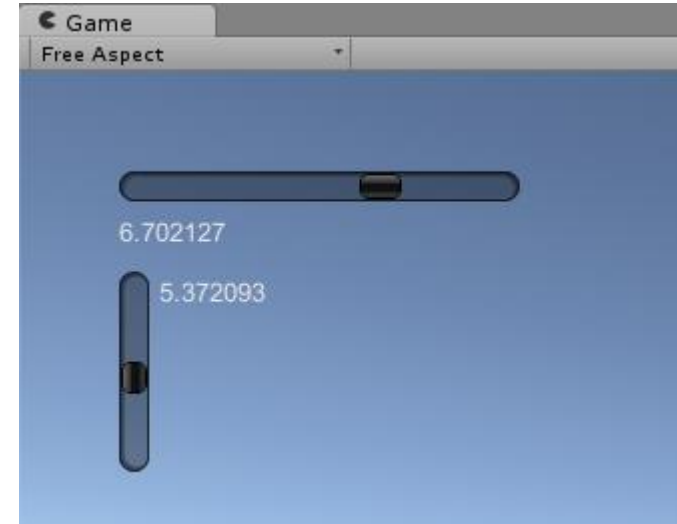
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUISlider : MonoBehaviour {
5
6     private float hSliderValue = 0.0f;
7     private float vSliderValue = 0.0f;
8
9     void OnGUI() {
10
11         hSliderValue = GUI.HorizontalSlider (new Rect (50, 50, 200, 20), hSliderValue, 0.0f, 10.0f);
12         GUI.Label (new Rect (50, 70, 100, 30), hSliderValue.ToString());
13
14         vSliderValue = GUI.VerticalSlider (new Rect (50, 100, 20, 100), vSliderValue, 0.0f, 10.0f);
15         GUI.Label (new Rect (70, 100, 100, 30), vSliderValue.ToString());
16     }
17 }
```



스크롤바

- 값을 조정하는데 이용하는 스크롤 바

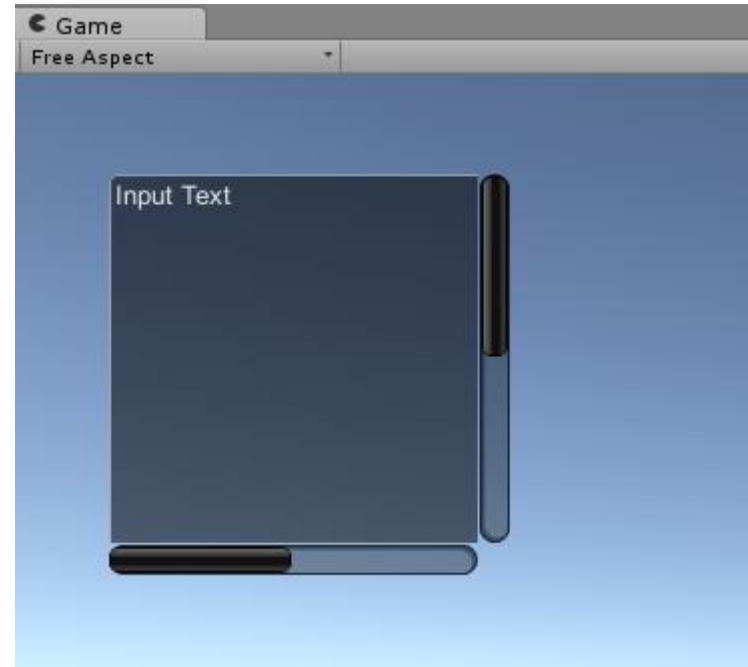
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIScrollBar : MonoBehaviour {
5
6     private float hScrollBarValue = 0.0f;
7     private float vScrollBarValue = 0.0f;
8
9     void OnGUI() {
10
11         hScrollBarValue = GUI.HorizontalScrollbar (new Rect (50, 50, 200, 20), hScrollBarValue, 0.5f, 0.0f, 10.5f);
12         GUI.Label (new Rect (50, 70, 100, 30), hScrollBarValue.ToString());
13
14         vScrollBarValue = GUI.VerticalScrollbar (new Rect (50, 100, 20, 100), vScrollBarValue, 0.5f, 0.0f, 10.5f);
15         GUI.Label (new Rect (70, 100, 100, 30), vScrollBarValue.ToString());
16     }
17 }
```



스크롤뷰

- 여러 UI컴포넌트들을 일정 영역내부에 표시

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIScrollView : MonoBehaviour {
5
6     public Vector2 scrollViewVector = Vector2.zero;
7     public string innerText = "Input Text";
8
9     void OnGUI() {
10         scrollViewVector = GUI.BeginScrollView (new Rect (50, 50, 200, 200),
11                                                 scrollViewVector,
12                                                 new Rect (0, 0, 400, 400));
13
14         innerText = GUI.TextArea (new Rect (0, 0, 400, 400), innerText);
15
16         GUI.EndScrollView ();
17     }
18 }
```



GUIContent

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIImage : MonoBehaviour {
5
6     public Texture2D thumbnail;
7
8     void OnGUI() {
9         GUI.Box (new Rect (50, 50, 100, 100), "thumbnail");
10        GUI.Box (new Rect (50, 250, 100, 100), thumbnail);
11        GUI.Box (new Rect (250, 50, 100, 100), new GUIContent("thumbnail", thumbnail));
12
13        GUI.Box (new Rect (250, 250, 100, 100), new GUIContent(thumbnail, "This is the tooltip"));
14        GUI.Label(new Rect(250, 350, 200, 30), GUI.tooltip);
15    }
16 }
```



GUI 디자인 변경

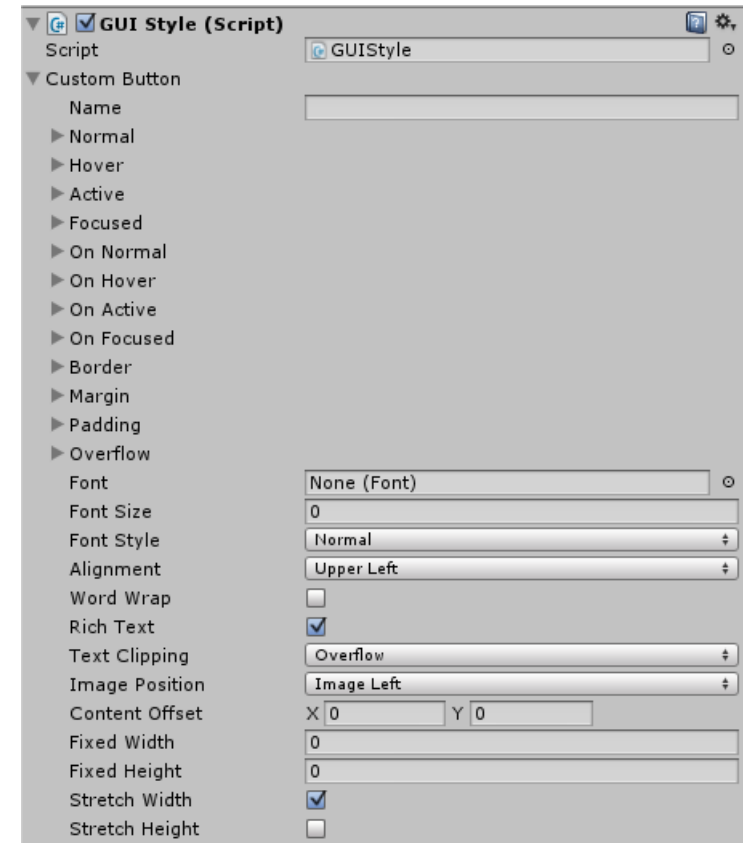
- 콘텐츠 고유의 GUI를 적용하기 위하여
 - GUIStyle
 - GUISkin



GUI Style

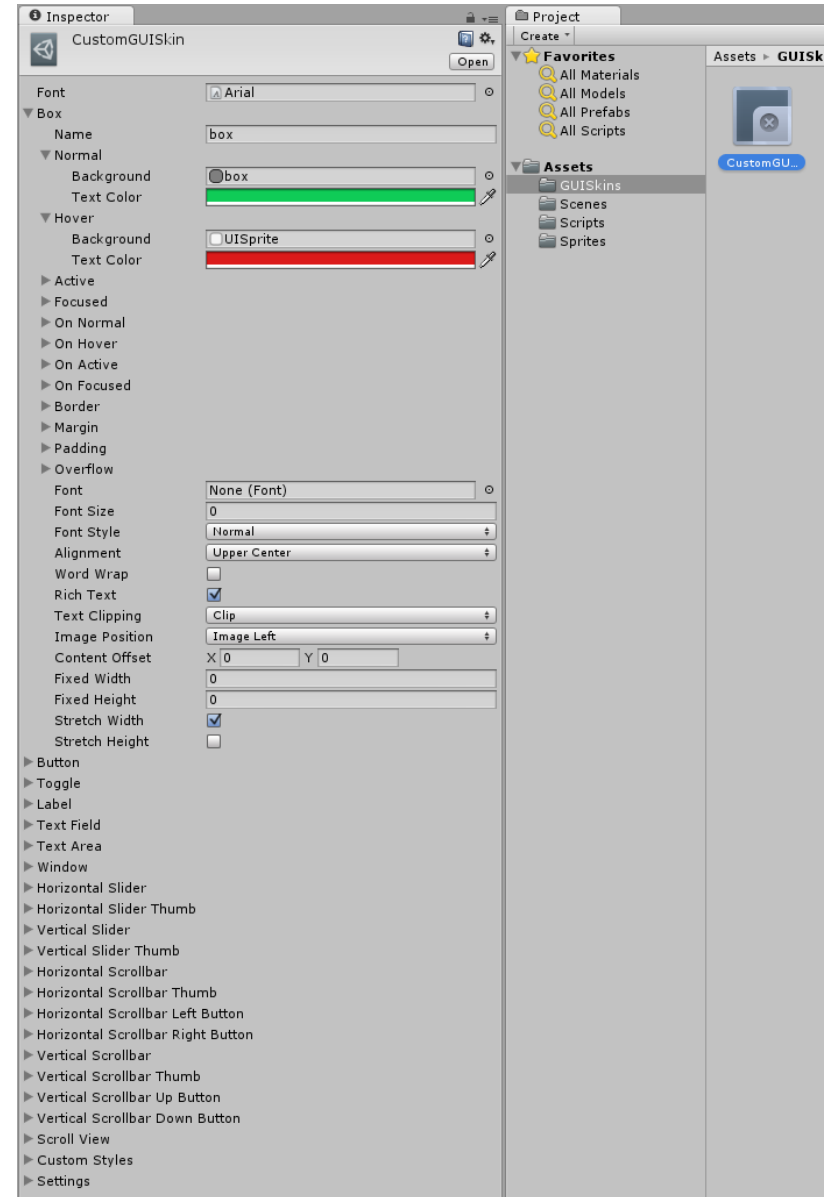
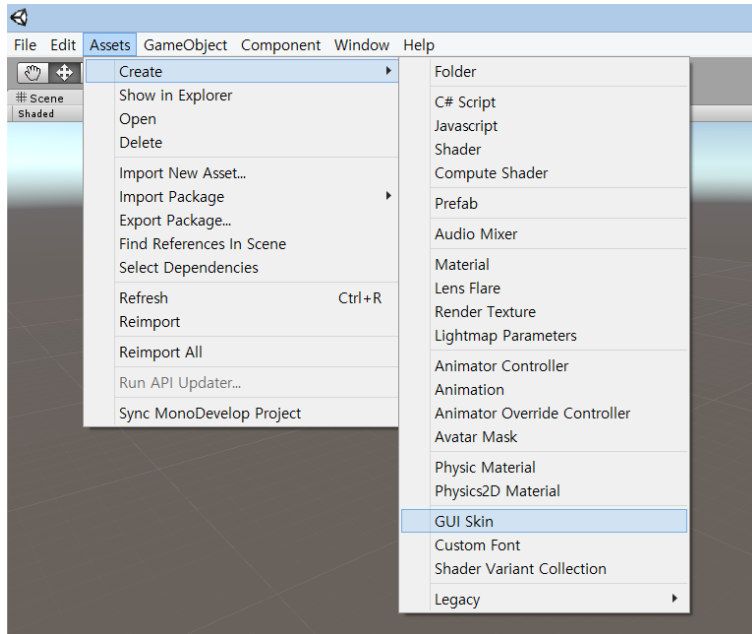
- GUI 스타일 확인하기
 - public 멤버 변수로 선언

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class CustomStyleBox : MonoBehaviour {
5
6     public UnityEngine.GUIStyle customStyle;
7
8     void OnGUI() {
9         GUI.Box (new Rect (100, 100, 175, 70), "Custom Style Box", customStyle);
10    }
11 }
```



GUI Skin

- UI 컨트롤들의 GUIStyle을 정의하는 애셋
- [메뉴 > Assets > Create > GUI Skin]



GUI Skin

- GUI Skin의 적용

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class CustomSkin : MonoBehaviour {
5     public GUISkin customSkin;
6
7     private bool toggle = true;
8
9     void OnGUI () {
10         GUI.skin = customSkin;
11         GUI.Box (new Rect(100, 250, 200, 20), "Custom GUISkin Box");
12
13         GUI.skin = null;
14         GUI.Box (new Rect(100, 300, 200, 20), "Default Box");
15     }
16 }
```



GUI를 배치하는 방법

- 고정 레이아웃 모드

- 직접 GUI 값을 설정

```
void OnGUI() {  
    GUI.Box (new Rect (100, 100, 175, 70), "Custom Style Box", customStyle);  
}
```

- 화면 크기 고려

- 사전에 디자인된 인터페이스가 있을 경우 사용

- 자동 레이아웃 모드

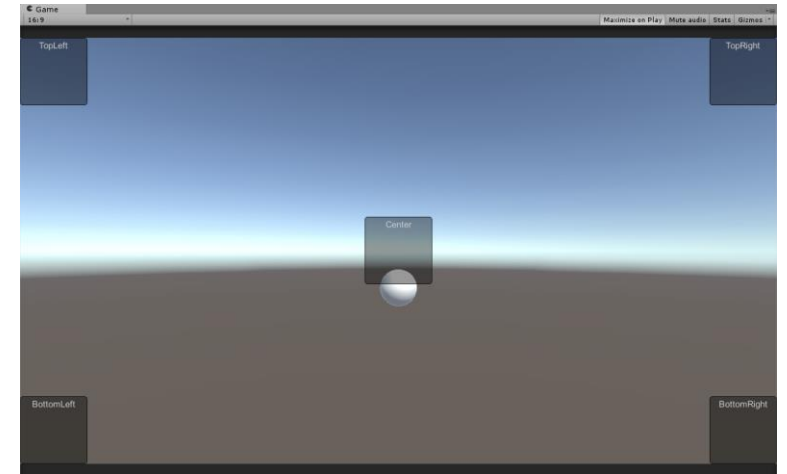
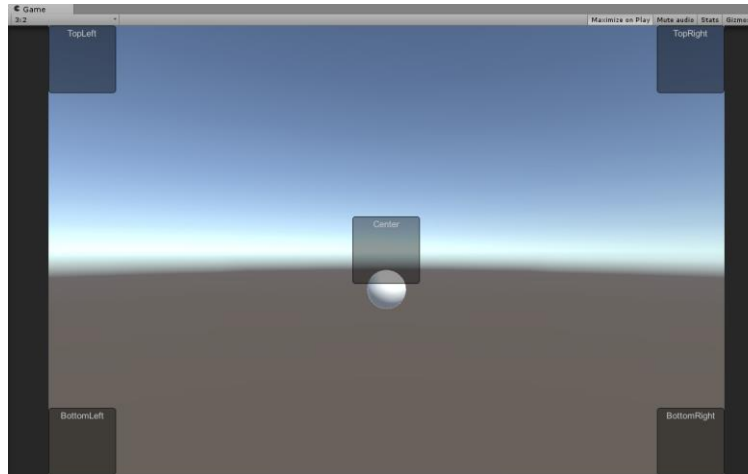
- GUILayout

- GUI의 개수와 배치가 변경될 경우 사용

Screen 클래스

- Screen 클래스
 - 화면과 관련된 정보를 담고 있음
 - 주요 멤버 변수
 - Screen.width - 화면 넓이
 - Screen.height - 화면 높이
 - Screen.orientation - 화면 오리엔테이션
- 일반적으로 GUI요소들은 화면 크기를 기준으로 위치를 설정

GUI의 배치



```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUIRelative : MonoBehaviour {
5     void OnGUI() {
6         GUI.Box (new Rect (0, 0, 100, 100), "TopLeft");
7         GUI.Box (new Rect (Screen.width - 100, 0, 100, 100), "TopRight");
8         GUI.Box (new Rect (0, Screen.height - 100, 100, 100), "BottomLeft");
9         GUI.Box (new Rect (Screen.width - 100, Screen.height - 100, 100, 100), "BottomRight");
10        GUI.Box (new Rect ((Screen.width - 100) / 2, (Screen.height - 100) / 2, 100, 100), "Center");
11    }
12 }
```

GUILayout

- GUI의 자동 레이아웃을 지원하는 클래스
- GUI의 위치를 설정하지 않는다.



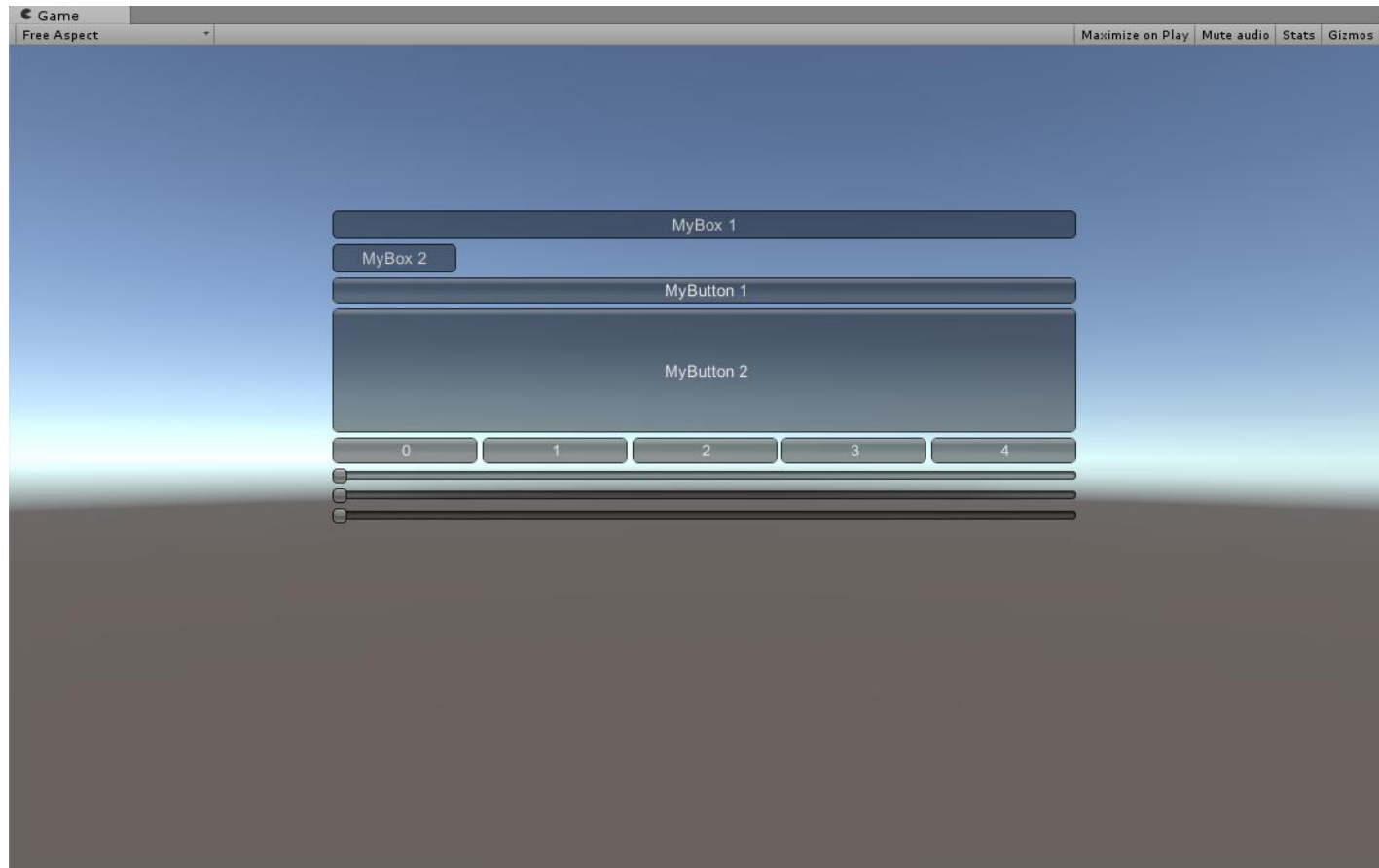
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUILayoutBox : MonoBehaviour {
5     void OnGUI () {
6         GUI.Box (new Rect(0, 50, 200, 20), "Manual Layout Box");
7
8         GUILayout.Box ("Auto Layout Box");
9     }
10 }
```

GUILayout

- 레이아웃 위치를 설정
- 레이아웃에 들어갈 GUI들을 배치
- 옵션 통해 GUI의 크기를 설정
- 수평 그룹
 - GUILayout.BeginHorizontal ();
 - GUILayout.EndHorizontal ();
- 수직 그룹
 - GUILayout.BeginVertical();
 - GUILayout.EndVertical();

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUILayoutEx : MonoBehaviour {
5     private float[] sliderValue = new float[3];
6
7     void OnGUI() {
8         GUILayout.BeginArea (new Rect ((Screen.width-600) / 2, (Screen.height-400) / 2, 600, 400));
9
10        GUILayout.Box ("MyBox 1");
11        GUILayout.Box ("MyBox 2", GUILayout.Width(100));
12        GUILayout.Button ("MyButton 1");
13        GUILayout.Button ("MyButton 2", GUILayout.Height (100));
14
15        GUILayout.BeginHorizontal ();
16        for (int i = 0; i < 5; i++) {
17            if (GUILayout.Button(i.ToString())) {
18                Debug.Log(i);
19            }
20        }
21        GUILayout.EndHorizontal ();
22
23        GUILayout.BeginVertical();
24        for (int j = 0; j < 3; j++) {
25            sliderValue[j] = GUILayout.HorizontalSlider(sliderValue[j], 0.0f, 10.0f);
26        }
27        GUILayout.EndVertical();
28
29        GUILayout.EndArea ();
30    }
31 }
```

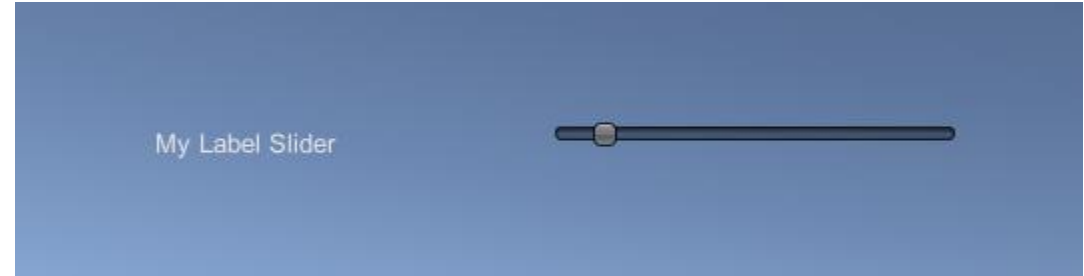

GUILayout



GUI 조합하기

- 여러개의 GUI가 함께 사용될 경우

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class GUILabelSlider : MonoBehaviour {
5
6     private float labelSlider = 1.0f;
7
8     void OnGUI() {
9         labelSlider = LabelSlider (new Rect (100, 100, 200, 50), labelSlider, 10.0f, "My Label Slider");
10    }
11
12    float LabelSlider(Rect rect, float value, float max, string text) {
13        GUI.Label (rect, text);
14
15        rect.x += rect.width;
16        value = GUI.HorizontalSlider (rect, value, 0.0f, max);
17        return value;
18    }
19 }
```



Unity UI System

- 유니티에서 UI들을 빠르고 직관적으로 제작하기 위한 시스템

