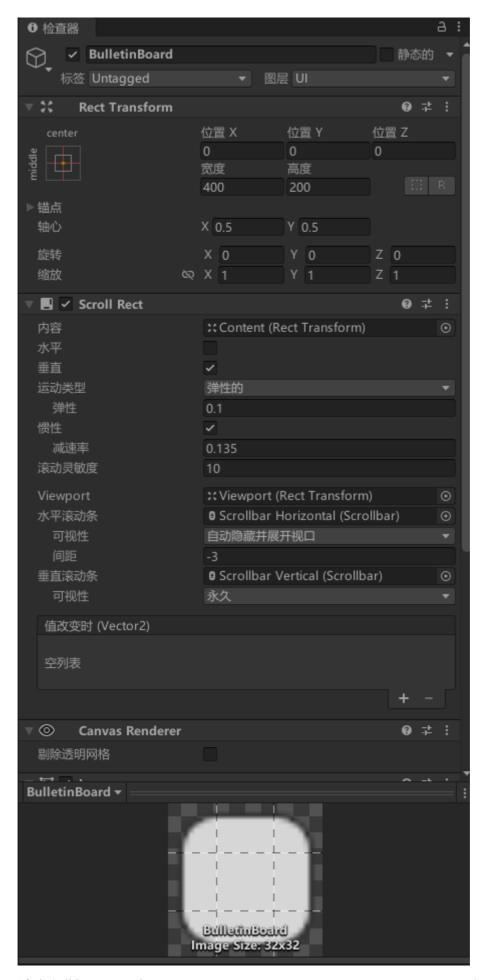
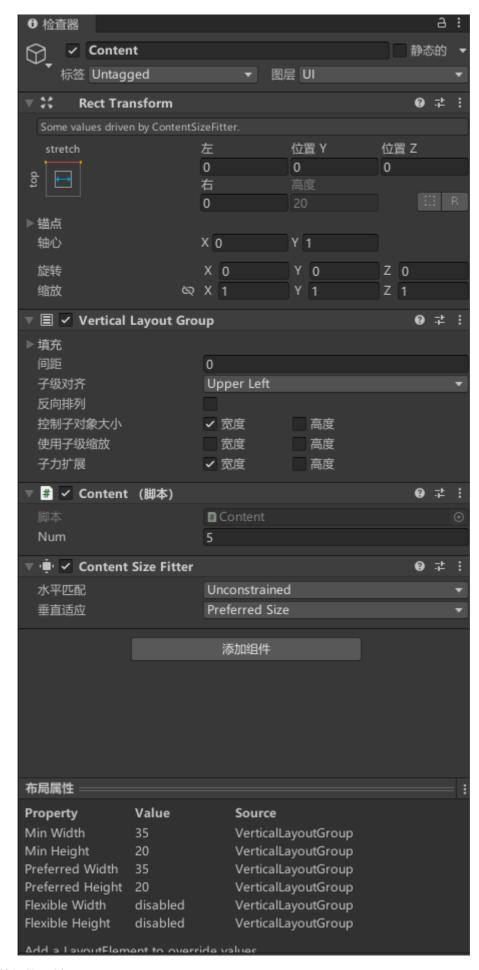
## UI 效果制作

- •进入 NGUI 官方网站,使用 UGUI 实现以下效果
- Quest Log 公告牌
- •以上例子需要使用 Unity web player, 仅支持以下操作系统与浏览器,参见官方下载
- Windows 版 IE11
- Mac OS X 10.7 Safari
- 出现界面需要等待较长时间, 打开页面让它慢慢加载

在Scene场景中添加ScrollView游戏对象,设置属性:取消了滚动视图的水平滚动,设置竖直滚动条永远可见。



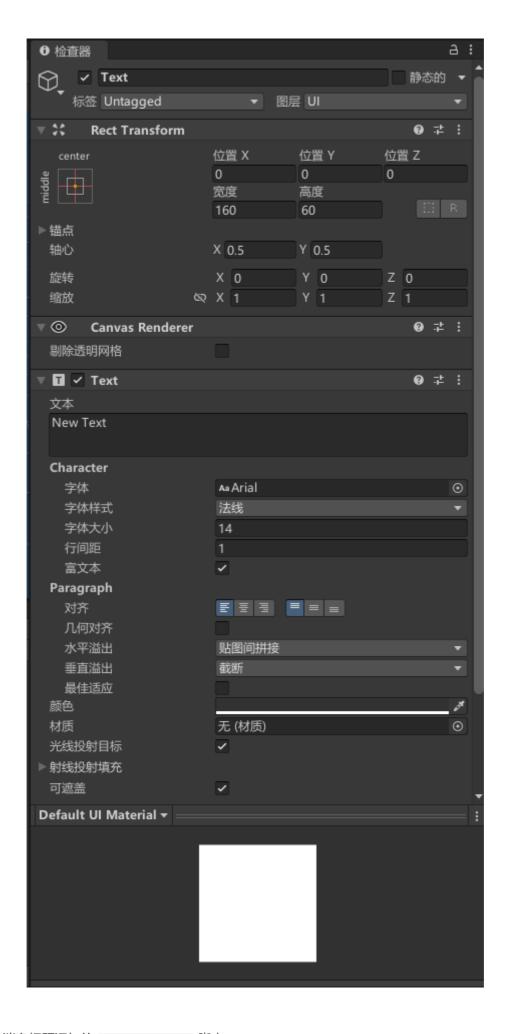
对Content列表内容进行设置:添加 Verticle Layout Group 和 Content Size Fitter 组件,前者用于构建垂直列表视图,后者用于实现列表高度的自适应。



消息标题按钮的预制:



消息主体内容的预制:



Button 消息标题添加的 ButtonHandler 脚本:

```
1 using System.Collections;
2
   using UnityEngine;
3
   using UnityEngine.UI;
4
5
   public class ButtonHandler: MonoBehaviour
6
7
       // 消息主体。
8
       public Text text;
9
       // 动画帧率。
10
       public int frame = 5;
       // 消息主体 Text 的高度。
11
12
       private float textHeight;
13
       // 消息主体 Text 的宽度。
       private float textWidth;
14
15
       void Start()
16
17
       {
           // 绑定响应点击事件。
18
           GetComponent<Button>().onClick.AddListener(OnClick);
19
20
           // 记录消息主体 Text 的高度。
           textHeight = text.rectTransform.sizeDelta.y;
21
22
           // 记录消息主体 Text 的宽度。
23
           textWidth = text.rectTransform.sizeDelta.x;
       }
24
25
       // 响应点击事件。
26
27
       void OnClick()
28
       {
           // 当消息主体已被展开,播放关闭动画。
29
30
           if (text.gameObject.activeSelf)
31
32
               StartCoroutine("CloseText");
33
           }
34
           else // 当消息主体未被展开,播放展开动画。
35
36
               StartCoroutine("OpenText");
37
38
       }
39
40
       // 播放关闭消息主体动画。
41
       private IEnumerator CloseText()
42
       {
           // 设置旋转角度的初始值和旋转速度。
43
44
           float angleX = 0;
45
           float angleSpeed = 90f / frame;
           // 设置 Text 初始高度和高度缩放速度。
46
47
           float height = textHeight;
48
           float heightSpeed = textHeight / frame;
49
50
           // 执行动画。
           for (int i = 0; i < frame; ++i)
51
52
           {
               // 更新旋转角度。
53
               anglex -= angleSpeed;
54
55
               // 更新 Text 高度。
56
               height -= heightSpeed;
57
               // 应用新的旋转角度和高度。
58
               text.transform.rotation = Quaternion.Euler(angleX, 0, 0);
```

```
text.rectTransform.sizeDelta = new Vector2(textWidth, height);
59
60
                // 结束动画。
                if (i == frame - 1)
61
62
63
                    text.gameObject.SetActive(false);
64
65
                yield return null;
66
            }
        }
67
68
69
        // 播放打开消息主体动画。
70
        private IEnumerator OpenText()
71
            // 设置旋转的初始值和旋转速度。
72
73
            float anglex = -90f;
            float angleSpeed = 90f / frame;
74
75
            // 设置 Text 初始高度和高度缩放速度。
            float height = 0;
76
            float heightSpeed = textHeight / frame;
77
78
            // 执行动画。
79
            for (int i = 0; i < frame; ++i)
80
81
            {
82
                // 更新旋转角度。
83
                anglex += angleSpeed;
84
                // 更新 Text 高度。
                height += heightSpeed;
85
                // 应用新的旋转角度和高度。
86
87
                text.transform.rotation = Quaternion.Euler(anglex, 0, 0);
88
                text.rectTransform.sizeDelta = new Vector2(textWidth, height);
89
                // 结束动画。
90
                if (i == 0)
91
92
                    text.gameObject.SetActive(true);
93
94
                yield return null;
95
96
        }
97
    }
```

## Content 添加的 Content.cs 脚本:

```
using UnityEngine;
 1
 2
    using UnityEngine.UI;
 3
    public class Content : MonoBehaviour
 4
 5
 6
        // num 指消息总数。
 7
        public int num = 10;
 8
        // 列表布局。
 9
        private RectTransform content;
10
11
        void Start()
12
        {
13
            content = GetComponent<RectTransform>();
            // 加载列表资源。
14
15
            LoadResources();
```

```
16
17
18
        // 加载列表资源。
        void LoadResources()
19
20
            for (int i = 0; i < num; ++i)
21
22
            {
                // 设置消息标题。
23
24
                Button title = Instantiate(Resources.Load<Button>
    ("Prefabs/Button"));
                title.name = "标题 " + (i + 1);
25
26
                title.GetComponentInChildren<Text>().text = "Message Title " +
    (i + 1);
27
                // 添加 Button 至消息列表。
                title.transform.SetParent(transform, false);
28
29
30
                // 设置消息主体。
                Text body = Instantiate(Resources.Load<Text>("Prefabs/Text"));
31
32
                body.name = "\pm 4" + (i + 1);
                body.text = "消息 " + (i + 1) + ".";
33
                // 添加 Text 至消息列表。
34
35
                body.transform.SetParent(transform, false);
36
                body.gameObject.SetActive(false);
37
                // 设置 Button 响应点击事件。
38
                ButtonHandler handler =
39
    title.gameObject.AddComponent<ButtonHandler>();
40
                handler.text = body;
41
            }
42
        }
43
```

## 最终呈现效果:

