

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 `Vertical 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
```

```
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;
11    // 消息主体 Text 的高度。
12    private float textHeight;
13    // 消息主体 Text 的宽度。
14    private float textwidth;
15
16    void Start()
17    {
18        // 绑定响应点击事件。
19        GetComponent<Button>().onClick.AddListener(OnClick);
20        // 记录消息主体 Text 的高度。
21        textHeight = text.rectTransform.sizeDelta.y;
22        // 记录消息主体 Text 的宽度。
23        textwidth = text.rectTransform.sizeDelta.x;
24    }
25
26    // 响应点击事件。
27    void onClick()
28    {
29        // 当消息主体已被展开，播放关闭动画。
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    {
43        // 设置旋转角度的初始值和旋转速度。
44        float angleX = 0;
45        float angleSpeed = 90f / frame;
46        // 设置 Text 初始高度和高度缩放速度。
47        float height = textHeight;
48        float heightSpeed = textHeight / frame;
49
50        // 执行动画。
51        for (int i = 0; i < frame; ++i)
52        {
53            // 更新旋转角度。
54            angleX -= angleSpeed;
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);
```

```

59         text.rectTransform.sizeDelta = new Vector2(textwidth, height);
60         // 结束动画。
61         if (i == frame - 1)
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 {
72     // 设置旋转的初始值和旋转速度。
73     float angleX = -90f;
74     float anglespeed = 90f / frame;
75     // 设置 Text 初始高度和高度缩放速度。
76     float height = 0;
77     float heightspeed = textHeight / frame;
78
79     // 执行动画。
80     for (int i = 0; i < frame; ++i)
81     {
82         // 更新旋转角度。
83         angleX += anglespeed;
84         // 更新 Text 高度。
85         height += heightspeed;
86         // 应用新的旋转角度和高度。
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:

```

1  using UnityEngine;
2  using UnityEngine.UI;
3
4  public class Content : MonoBehaviour
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>();
14         // 加载列表资源。
15         LoadResources();

```

```

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

```

最终呈现效果：

