

Demo7-发起Web请求

2018年3月27日 7:43

虽然官方目前还没有写WebRequest组件的文档，但是，这个组件的API相对比较简单，我也简单地演示其中一个吧。

1. API列表

int	AddWebRequest (string webRequestUri)	增加 Web 请求任务。 更多...
int	AddWebRequest (string webRequestUri, byte[] postData)	增加 Web 请求任务。 更多...
int	AddWebRequest (string webRequestUri, WWWForm wwwForm)	增加 Web 请求任务。 更多...
int	AddWebRequest (string webRequestUri, object userData)	增加 Web 请求任务。 更多...
int	AddWebRequest (string webRequestUri, byte[] postData, object userData)	增加 Web 请求任务。 更多...
int	AddWebRequest (string webRequestUri, WWWForm wwwForm, object userData)	增加 Web 请求任务。 更多...
bool	RemoveWebRequest (int serialId)	移除 Web 请求任务。 更多...
void	RemoveAllWebRequests ()	移除所有 Web 请求任务。 更多...

WebRequest组件用于发起短链接的请求，包括Get、Post，以上几个不同的AddWebRequest函数是用于在发生数据的时候附上post数据、Form表单数据，这个看参数名就知道了。至于userData，当然是自定义的数据，在请求成功或失败时可以获取这个自定义数据。

2. 一个简单的Get请求

新建一个场景 (Demo7_Launch) 和一个初始流程 (Demo7_ProcedureLaunch)，流程代码如下：

```
using GameFramework;
using GameFramework.Event;
using GameFramework.Procedure;
using UnityEngine;
using UnityGameFramework.Runtime;
using ProcedureOwner = GameFramework.Fsm.IFsm<GameFramework.Procedure.IProcedureManager>;
public class Demo7_ProcedureLaunch : ProcedureBase {
    protected override void OnEnter (ProcedureOwner procedureOwner) {
        base.OnEnter (procedureOwner);
        // 获取框架事件组件
        EventComponent Event
            = UnityGameFramework.Runtime.GameEntry.GetComponent<EventComponent> ();
        Event.Subscribe (WebRequestSuccessEventArgs.EventId, OnWebRequestSuccess);
        Event.Subscribe (WebRequestFailureEventArgs.EventId, OnWebRequestFailure);
        // 获取框架网络组件
        WebRequestComponent WebRequest
            = UnityGameFramework.Runtime.GameEntry.GetComponent<WebRequestComponent> ();
        string url = "http://gameframework.cn/starforce/version.txt";
        WebRequest.AddWebRequest (url, this);
    }
    private void OnWebRequestSuccess (object sender, GameEventArgs e) {
        WebRequestSuccessEventArgs ne = (WebRequestSuccessEventArgs) e;
        // 获取回应的数据
        string responseJson = Utility.Converter.GetString (ne.GetWebResponseBytes ());
        Log.Debug ("responseJson: " + responseJson);
    }
}
```

```

    }
    private void OnWebRequestFailure (object sender, GameEventArgs e) {
        Log.Warning ("请求失败");
    }
}

```

使用Event组件订阅请求成功和失败事件，使用WebRequest组件的AddWebRequest发起请求，请求的地址是官方博客的一个文本文件。

设置好初始流程后，运行游戏，如果你能看到以下日志输出，那就代表你成功了：

