**Unity3D中C#使用StreamReader类读取文件**

Posted on 2013年03月25日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 281 次

[**Unity3D**](http://www.unitymanual.com)中C#使用StreamReader类读取文件文件。在很多情况下，数据源是磁盘上的文件或网络的某些位置。任何可以发送数据的位置都可以是数据源，比如网络应用程序、Web服务，甚至是控制台。用来从文件中读取数据的类是StreamReader。同StreamWriter一样，这是一个通用类，可以用于任何流。StreamReader对象的创建方式非常类似于StreamWriter对象。创建它的最常见方式是使用前面创建的FileStream对象：

相关文章：[**Unity3D中C#使用StreamWriter类写入文件**](http://www.unitymanual.com/3219.html)

FileStream fs = new FileStream("test.txt",FileMode.Open);

StreamReader sr = new StreamReader(fs);

同StreamWriter一样，StreamReader类可以直接在包含具体文件路径的字符串中创建：

StreamReader sr = new StreamReader("test.txt");

using System;

using System.Collections.Generic;

using System.Text;

using System.IO;

class Program

{

static void Main(string[]args)

{

try

{

FileStream aFile = new FileStream(@"c:\祝福.txt", FileMode.Open);

StreamReader sr = new StreamReader(aFile);

string strLine = sr.ReadLine();

while(strLine != null)

{

Console.WriteLine(strLine);

strLine = sr.ReadLine();

}

sr.Close();

}

catch (IOException ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

return ;

}

Console.ReadKey();

}

}

FileStream构造函数找不到该文件，就会抛出异常。同前面一样，必须导入System.IO命名空间，才能访问需要的类。

使用ReadLine()方法从文件中读取文本。这个方法读取回车符之前的文本，并以字符串的形式返回结果文本。当到达文件尾时，该方法就返回空值，通过这种方法可以测试文件是否已到达了尾部。注意使用while循环时，在执行循环体的代码之前进行检查，确保读取的行不为空，这样就只显示文件的有效内容：

strLine = sr.ReadLine();

while(strLine != null)

{

Console.WriteLine(strLine);

strLine = sr.ReadLine();

}

ReadLine()方法不是在文件中访问数据的惟一方法。StreamReader类还有许多读取数据的方法。

读取数据最简单的方法是Read()。此方法将流的下一个字符作为正整数值返回，如果到达了流的结尾处，则返回–1。使用Convert实用类可以把这个值转换为字符。在上面的示例中，程序的主体可以按如下方式编写：

StreamReader sr = new StreamReader(aFile);

int nChar;

nChar = sr.Read();

while(nChar != -1)

{

Console.Write(Convert.ToChar(nChar));

nChar = sr.Read();

}

sr.Close();

对于小型文件，可以使用一个非常方便的方法，即ReadToEnd()方法。此方法读取整个文件，并将其作为字符串返回。在此，前面的应用程序可以简化为：

StreamReader sr = new StreamReader(aFile);

strLine = sr.ReadToEnd();

Console.WriteLine(strLine);

sr.Close();

这似乎非常容易和方便，但必须小心。将所有的数据读取到字符串对象中，会迫使文件中的数据放到内存中。应根据数据文件的大小禁止这样处理。如果数据文件非常大，最好将数据留在文件中，并使用StreamReader的方法访问文件。