**Unity3D读写XML**

Posted on 2013年07月11日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 17 次

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 001 | | | **using** UnityEngine; | |
| 002 | | | **using** System.Collections; | |
| 003 | | | **using** System.Xml; | |
| 004 | | | **using** System.Xml.Serialization; | |
| 005 | | | **using** System.IO; | |
| 006 | | | **using** System.Text; | |
| 007 | | |  | |
| 008 | | | **public** **class** \_GameSaveLoad: MonoBehaviour { | |
| 009 | | |  | |
| 010 | | | *// An example where the encoding can be found is at* | |
| 011 | | | *// http://www.eggheadcafe.com/articles/system.xml.xmlserialization.asp* | |
| 012 | | | *// We will just use the KISS method and cheat a little and use* | |
| 013 | | | *// the examples from the web page since they are fully described* | |
| 014 | | |  | |
| 015 | | | *// This is our local private members* | |
| 016 | | | Rect \_Save, \_Load, \_SaveMSG, \_LoadMSG; | |
| 017 | | | **bool** \_ShouldSave, \_ShouldLoad,\_SwitchSave,\_SwitchLoad; | |
| 018 | | | **string** \_FileLocation,\_FileName; | |
| 019 | | | **public** GameObject \_Player; | |
| 020 | | | UserData myData; | |
| 021 | | | **string** \_PlayerName; | |
| 022 | | | **string** \_data; | |
| 023 | | |  | |
| 024 | | | Vector3 VPosition; | |
| 025 | | |  | |
| 026 | | | *// When the EGO is instansiated the Start will trigger* | |
| 027 | | | *// so we setup our initial values for our local members* | |
| 028 | | | **void** Start () { | |
| 029 | | | *// We setup our rectangles for our messages* | |
| 030 | | | \_Save=new Rect(10,80,100,20); | |
| 031 | | | \_Load=new Rect(10,100,100,20); | |
| 032 | | | \_SaveMSG=new Rect(10,120,400,40); | |
| 033 | | | \_LoadMSG=new Rect(10,140,400,40); | |
| 034 | | |  | |
| 035 | | | *// Where we want to save and load to and from* | |
| 036 | | | \_FileLocation=Application.dataPath; | |
| 037 | | | \_FileName="SaveData.xml"; | |
| 038 | | |  | |
| 039 | | | *// for now, lets just set the name to Joe Schmoe* | |
| 040 | | | \_PlayerName = "Joe Schmoe"; | |
| 041 | | |  | |
| 042 | | | *// we need soemthing to store the information into* | |
| 043 | | | myData=new UserData(); | |
| 044 | | | } | |
| 045 | | |  | |
| 046 | | | **void** Update () {} | |
| 047 | | |  | |
| 048 | | | **void** OnGUI() | |
| 049 | | | { | |
| 050 | | |  | |
| 051 | | | *//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** | |
| 052 | | | *// Loading The Player...* | |
| 053 | | | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** | |
| 054 | | | **if** (GUI.Button(\_Load,"Load")) { | |
| 055 | | |  | |
| 056 | | | GUI.Label(\_LoadMSG,"Loading from: "+\_FileLocation); | |
| 057 | | | *// Load our UserData into myData* | |
| 058 | | | LoadXML(); | |
| 059 | | | **if**(\_data.ToString() != "") | |
| 060 | | | { | |
| 061 | | | *// notice how I use a reference to type (UserData) here, you need this* | |
| 062 | | | *// so that the returned object is converted into the correct type* | |
| 063 | | | myData = (UserData)DeserializeObject(\_data); | |
| 064 | | | *// set the players position to the data we loaded* | |
| 065 | | | VPosition=new Vector3(myData.\_iUser.x,myData.\_iUser.y,myData.\_iUser.z); | |
| 066 | | | \_Player.transform.position=VPosition; | |
| 067 | | | *// just a way to show that we loaded in ok* | |
| 068 | | | Debug.Log(myData.\_iUser.name); | |
| 069 | | | } | |
| 070 | | |  | |
| 071 | | | } | |
| 072 | | | *//Unity3D教程手册：www.unitymanual.com* | |
| 073 | | | *//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** | |
| 074 | | | *// Saving The Player...* | |
| 075 | | | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** | |
| 076 | | | **if** (GUI.Button(\_Save,"Save")) { | |
| 077 | | |  | |
| 078 | | | GUI.Label(\_SaveMSG,"Saving to: "+\_FileLocation); | |
| 079 | | | myData.\_iUser.x=\_Player.transform.position.x; | |
| 080 | | | myData.\_iUser.y=\_Player.transform.position.y; | |
| 081 | | | myData.\_iUser.z=\_Player.transform.position.z; | |
| 082 | | | myData.\_iUser.name=\_PlayerName; | |
| 083 | | |  | |
| 084 | | | *// Time to creat our XML!* | |
| 085 | | | \_data = SerializeObject(myData); | |
| 086 | | | *// This is the final resulting XML from the serialization process* | |
| 087 | | | CreateXML(); | |
| 088 | | | Debug.Log(\_data); | |
| 089 | | | } | |
| 090 | | |  | |
| 091 | | |  | |
| 092 | | | } | |
| 093 | | |  | |
| 094 | | | */\* The following metods came from the referenced URL \*/* | |
| 095 | | | **string** UTF8ByteArrayToString(**byte**[] characters) | |
| 096 | | | { | |
| 097 | | | UTF8Encoding encoding = new UTF8Encoding(); | |
| 098 | | | **string** constructedString = encoding.GetString(characters); | |
| 099 | | | **return** (constructedString); | |
| 100 | | | } | |
| 101 | | |  | |
| 102 | | | **byte**[] StringToUTF8ByteArray(**string** pXmlString) | |
| 103 | | | { | |
| 104 | | | UTF8Encoding encoding = new UTF8Encoding(); | |
| 105 | | | **byte**[] byteArray = encoding.GetBytes(pXmlString); | |
| 106 | | | **return** byteArray; | |
| 107 | | | } | |
| 108 | | |  | |
| 109 | | | *// Here we serialize our UserData object of myData* | |
| 110 | | | **string** SerializeObject(**object** pObject) | |
| 111 | | | { | |
| 112 | | | **string** XmlizedString = **null**; | |
| 113 | | | MemoryStream memoryStream = new MemoryStream(); | |
| 114 | | | XmlSerializer xs = new XmlSerializer(typeof(UserData)); | |
| 115 | | | XmlTextWriter xmlTextWriter = new XmlTextWriter(memoryStream, Encoding.UTF8); | |
| 116 | | | xs.Serialize(xmlTextWriter, pObject); | |
| 117 | | | memoryStream = (MemoryStream)xmlTextWriter.BaseStream; | |
| 118 | | | XmlizedString = UTF8ByteArrayToString(memoryStream.ToArray()); | |
| 119 | | | **return** XmlizedString; | |
| 120 | | | } | |
| 121 | | |  | |
| 122 | | | *// Here we deserialize it back into its original form* | |
| 123 | | | **object** DeserializeObject(**string** pXmlizedString) | |
| 124 | | | { | |
| 125 | | | XmlSerializer xs = new XmlSerializer(typeof(UserData)); | |
| 126 | | | MemoryStream memoryStream = new MemoryStream(StringToUTF8ByteArray(pXmlizedString)); | |
| 127 | | | XmlTextWriter xmlTextWriter = new XmlTextWriter(memoryStream, Encoding.UTF8); | |
| 128 | | | **return** xs.Deserialize(memoryStream); | |
| 129 | | | } | |
| 130 | | | *//Unity3D教程手册：www.unitymanual.com* | |
| 131 | | | *// Finally our save and load methods for the file itself* | |
| 132 | | | **void** CreateXML() | |
| 133 | | | { | |
| 134 | | | StreamWriter writer; | |
| 135 | | | FileInfo t = new FileInfo(\_FileLocation+"**\\**"+ \_FileName); | |
| 136 | | | **if**(!t.Exists) | |
| 137 | | | { | |
| 138 | | | writer = t.CreateText(); | |
| 139 | | | } | |
| 140 | | | **else** | |
| 141 | | | { | |
| 142 | | | t.Delete(); | |
| 143 | | | writer = t.CreateText(); | |
| 144 | | | } | |
| 145 | | | writer.Write(\_data); | |
| 146 | | | writer.Close(); | |
| 147 | | | Debug.Log("File written."); | |
| 148 | | | } | |
| 149 | | |  | |
| 150 | | | **void** LoadXML() | |
| 151 | | | { | |
| 152 | | | StreamReader r = File.OpenText(\_FileLocation+"**\\**"+ \_FileName); | |
| 153 | | | **string** \_info = r.ReadToEnd(); | |
| 154 | | | r.Close(); | |
| 155 | | | \_data=\_info; | |
| 156 | | | Debug.Log("File Read"); | |
| 157 | | | } | |
| 158 | | | } | |
| 159 | | |  | |
| 160 | | | *// UserData is our custom class that holds our defined objects we want to store in XML format* | |
| 161 | | | **public** **class** UserData | |
| 162 | | | { | |
| 163 | | | *// We have to define a default instance of the structure* | |
| 164 | | | **public** DemoData \_iUser; | |
| 165 | | | *// Default constructor doesn't really do anything at the moment* | |
| 166 | | | **public** UserData() { } | |
| 167 | | |  | |
| 168 | | | *// Anything we want to store in the XML file, we define it here* | |
| 169 | | | **public** **struct** DemoData | |
| 170 | | | { | |
| 171 | | | **public** **float** x; | |
| 172 | | | **public** **float** y; | |
| 173 | | | **public** **float** z; | |
| 174 | | | **public** **string** name; | |
| 175 | | | } | |
| 176 | | | } | |
| 177 | | |  | |
| 178 | | |  | |
|  |  | |  |

|  |  |
| --- | --- |
| 001 | import System; |
| 002 | import System.Collections; |
| 003 | import System.Xml; |
| 004 | import System.Xml.Serialization; |
| 005 | import System.IO; |
| 006 | import System.Text; |
| 007 |  |
| 008 | *// Anything we want to store in the XML file, we define it here* |
| 009 | **class** DemoData |
| 010 | { |
| 011 | **var** x : **float**; |
| 012 | **var** y : **float**; |
| 013 | **var** z : **float**; |
| 014 | **var** name : **String**; |
| 015 | } |
| 016 |  |
| 017 | *// UserData is our custom class that holds our defined objects we want to store in XML format* |
| 018 | **class** UserData |
| 019 | { |
| 020 | *// We have to define a default instance of the structure* |
| 021 | **public** **var** \_iUser : DemoData = new DemoData(); |
| 022 | *// Default constructor doesn't really do anything at the moment* |
| 023 | function UserData() { } |
| 024 | } |
| 025 |  |
| 026 | *//public class GameSaveLoad: MonoBehaviour {* |
| 027 |  |
| 028 | *// An example where the encoding can be found is at* |
| 029 | *// http://www.eggheadcafe.com/articles/system.xml.xmlserialization.asp* |
| 030 | *// We will just use the KISS method and cheat a little and use* |
| 031 | *// the examples from the web page since they are fully described* |
| 032 |  |
| 033 | *// This is our local private members* |
| 034 | **private** **var** \_Save : Rect; |
| 035 | **private** **var** \_Load : Rect; |
| 036 | **private** **var** \_SaveMSG : Rect; |
| 037 | **private** **var** \_LoadMSG : Rect; |
| 038 | *//var \_ShouldSave : boolean;* |
| 039 | *//var \_ShouldLoad : boolean;* |
| 040 | *//var \_SwitchSave : boolean;* |
| 041 | *//var \_SwitchLoad : boolean;* |
| 042 | **private** **var** \_FileLocation : **String**; |
| 043 | **private** **var** \_FileName : **String** = "SaveData.xml"; |
| 044 |  |
| 045 | *//public GameObject \_Player;* |
| 046 | **var** \_Player : GameObject; |
| 047 | **var** \_PlayerName : **String** = "Joe Schmoe"; |
| 048 |  |
| 049 | **private** **var** myData : UserData; |
| 050 | **private** **var** \_data : **String**; |
| 051 |  |
| 052 | **private** **var** VPosition : Vector3; |
| 053 |  |
| 054 | *// When the EGO is instansiated the Start will trigger* |
| 055 | *// so we setup our initial values for our local members* |
| 056 | *//function Start () {* |
| 057 | function Awake () { |
| 058 | *// We setup our rectangles for our messages* |
| 059 | \_Save=new Rect(10,80,100,20); |
| 060 | \_Load=new Rect(10,100,100,20); |
| 061 | \_SaveMSG=new Rect(10,120,200,40); |
| 062 | \_LoadMSG=new Rect(10,140,200,40); |
| 063 |  |
| 064 | *// Where we want to save and load to and from* |
| 065 | \_FileLocation=Application.dataPath; |
| 066 |  |
| 067 |  |
| 068 | *// we need soemthing to store the information into* |
| 069 | myData=new UserData(); |
| 070 | } |
| 071 |  |
| 072 | function Update () {} |
| 073 |  |
| 074 | function OnGUI() |
| 075 | { |
| 076 |  |
| 077 | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** |
| 078 | *// Loading The Player...* |
| 079 | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** |
| 080 | **if** (GUI.Button(\_Load,"Load")) { |
| 081 |  |
| 082 | GUI.Label(\_LoadMSG,"Loading from: "+\_FileLocation); |
| 083 | *// Load our UserData into myData* |
| 084 | LoadXML(); |
| 085 | **if**(\_data.ToString() != "") |
| 086 | { |
| 087 | *// notice how I use a reference to type (UserData) here, you need this* |
| 088 | *// so that the returned object is converted into the correct type* |
| 089 | *//myData = (UserData)DeserializeObject(\_data);* |
| 090 | myData = DeserializeObject(\_data); |
| 091 | *// set the players position to the data we loaded* |
| 092 | VPosition=new Vector3(myData.\_iUser.x,myData.\_iUser.y,myData.\_iUser.z); |
| 093 | \_Player.transform.position=VPosition; |
| 094 | *// just a way to show that we loaded in ok* |
| 095 | Debug.Log(myData.\_iUser.name); |
| 096 | } |
| 097 |  |
| 098 | } |
| 099 |  |
| 100 | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** |
| 101 | *// Saving The Player...* |
| 102 | *// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** |
| 103 | **if** (GUI.Button(\_Save,"Save")) { |
| 104 |  |
| 105 | GUI.Label(\_SaveMSG,"Saving to: "+\_FileLocation); |
| 106 | *//Debug.Log("SaveLoadXML: sanity check:"+ \_Player.transform.position.x);* |
| 107 |  |
| 108 | myData.\_iUser.x = \_Player.transform.position.x; |
| 109 | myData.\_iUser.y = \_Player.transform.position.y; |
| 110 | myData.\_iUser.z = \_Player.transform.position.z; |
| 111 | myData.\_iUser.name = \_PlayerName; |
| 112 |  |
| 113 | *// Time to creat our XML!* |
| 114 | \_data = SerializeObject(myData); |
| 115 | *// This is the final resulting XML from the serialization process* |
| 116 | CreateXML(); |
| 117 | Debug.Log(\_data); |
| 118 | } |
| 119 |  |
| 120 |  |
| 121 | } |
| 122 |  |
| 123 | */\* The following metods came from the referenced URL \*/* |
| 124 | *//string UTF8ByteArrayToString(byte[] characters)* |
| 125 | function UTF8ByteArrayToString(characters : **byte**[] ) |
| 126 | { |
| 127 | **var** encoding : UTF8Encoding = new UTF8Encoding(); |
| 128 | **var** constructedString : **String** = encoding.GetString(characters); |
| 129 | **return** (constructedString); |
| 130 | } |
| 131 |  |
| 132 | *//byte[] StringToUTF8ByteArray(string pXmlString)* |
| 133 | function StringToUTF8ByteArray(pXmlString : **String**) |
| 134 | { |
| 135 | **var** encoding : UTF8Encoding = new UTF8Encoding(); |
| 136 | **var** byteArray : **byte**[] = encoding.GetBytes(pXmlString); |
| 137 | **return** byteArray; |
| 138 | } |
| 139 |  |
| 140 | *// Here we serialize our UserData object of myData* |
| 141 | *//string SerializeObject(object pObject)* |
| 142 | function SerializeObject(pObject : **Object**) |
| 143 | { *//Unity3D教程手册：www.unitymanual.com* |
| 144 | **var** XmlizedString : **String** = **null**; |
| 145 | **var** memoryStream : MemoryStream = new MemoryStream(); |
| 146 | **var** xs : XmlSerializer = new XmlSerializer(typeof(UserData)); |
| 147 | **var** xmlTextWriter : System.Xml.XmlTextWriter = new System.Xml.XmlTextWriter(memoryStream, Encoding.UTF8); |
| 148 | xs.Serialize(xmlTextWriter, pObject); |
| 149 | memoryStream = xmlTextWriter.BaseStream; *// (MemoryStream)* |
| 150 | XmlizedString = UTF8ByteArrayToString(memoryStream.ToArray()); |
| 151 | **return** XmlizedString; |
| 152 | } |
| 153 |  |
| 154 | *// Here we deserialize it back into its original form* |
| 155 | *//object DeserializeObject(string pXmlizedString)* |
| 156 | function DeserializeObject(pXmlizedString : **String**) |
| 157 | { |
| 158 | **var** xs : XmlSerializer = new XmlSerializer(typeof(UserData)); |
| 159 | **var** memoryStream : MemoryStream = new MemoryStream(StringToUTF8ByteArray(pXmlizedString)); |
| 160 | **var** xmlTextWriter : System.Xml.XmlTextWriter = new System.Xml.XmlTextWriter(memoryStream, Encoding.UTF8); |
| 161 | **return** xs.Deserialize(memoryStream); |
| 162 | } |
| 163 | *//Unity3D教程手册：www.unitymanual.com* |
| 164 | *// Finally our save and load methods for the file itself* |
| 165 | function CreateXML() |
| 166 | { |
| 167 | **var** writer : StreamWriter; |
| 168 | *//FileInfo t = new FileInfo(\_FileLocation+"\\"+ \_FileName);* |
| 169 | **var** t : FileInfo = new FileInfo(\_FileLocation+"/"+ \_FileName); |
| 170 | **if**(!t.Exists) |
| 171 | { |
| 172 | writer = t.CreateText(); |
| 173 | } |
| 174 | **else** |
| 175 | { |
| 176 | t.Delete(); |
| 177 | writer = t.CreateText(); |
| 178 | } |
| 179 | writer.Write(\_data); |
| 180 | writer.Close(); |
| 181 | Debug.Log("File written."); |
| 182 | } |
| 183 |  |
| 184 | function LoadXML() |
| 185 | { |
| 186 | *//StreamReader r = File.OpenText(\_FileLocation+"\\"+ \_FileName);* |
| 187 | **var** r : StreamReader = File.OpenText(\_FileLocation+"/"+ \_FileName); |
| 188 | **var** \_info : **String** = r.ReadToEnd(); |
| 189 | r.Close(); |
| 190 | \_data=\_info; |
| 191 | Debug.Log("File Read"); |
| 192 | } |