**Unity3D教程：切割图片**

Posted on 2013年06月28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 103 次

切割图片：用Unity把图片的像素中不为alpha的部分切出来保存成单个图片。

之前打算用Texture2D.GetPixel()遍历整张图片，判断每一个像素的alpha是否为0，接着对每一个像素判断是否为临界的像素点，最后用new一个Texture2D用Texture2D.SetPixel()赋值像素的RGBA。但是存在一种特殊的情况，那就是想要截取的图片中有alpha=0的时候。于是乎又另图思路，结合CEImagesetEditor这个开源的工具来完成这个工作，虽然这种方法不够“智能”，但是截取的很准确。

用CEImagesetEditor工具定位要截取的位置和大小后能生成一个XML文件，这个文件包含要截取图片的名字、位置以及大小。XML文件形如：

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

|  |  |
| --- | --- |
| 1 | <?xml version="1.0" encoding="UTF-8"?> |
| 2 |  |
| 3 | <Imageset Name="test" Imagefile="test**\r**oomlist.tga" > |
| 4 | <Image Name="name" XPos="190" YPos="59" Width="60" Height="25" ></Image> |
| 5 | <Image Name="photo" XPos="32" YPos="48" Width="127" Height="206" ></Image> |
| 6 | <Image Name="sumbit" XPos="55" YPos="264" Width="65" Height="38" ></Image> |
| 7 | </Imageset> |

接着依次做的事情有：解析XML文件，新建Texture2D，根据XML的信息遍历制定的区域，获取指定区域的像素RGBA值，RGBA赋值给Texture2D，保存到当前工程的某一目录中。

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

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 |  |
| 003 | **using** System; |
| 004 |  |
| 005 | **using** System.Collections; |
| 006 |  |
| 007 | **using** System.Xml; |
| 008 |  |
| 009 | **using** System.IO; |
| 010 |  |
| 011 | **public** **class** ClipPic : MonoBehaviour { |
| 012 |  |
| 013 | TextureD newTexture; |
| 014 |  |
| 015 | **public** TextureD resTexture; |
| 016 |  |
| 017 | Color color; |
| 018 |  |
| 019 | **string** picName; |
| 020 |  |
| 021 | **int** picPos\_x,picPos\_y; |
| 022 |  |
| 023 | **int** picWidth,picHeight; |
| 024 |  |
| 025 | *// Use this for initialization* |
| 026 |  |
| 027 | **void** Start () { |
| 028 |  |
| 029 | XmlDocument xmlDoc = new XmlDocument(); |
| 030 |  |
| 031 | **string** xml = Resources.Load(“test”).ToString(); |
| 032 |  |
| 033 | xmlDoc.LoadXml(xml); |
| 034 |  |
| 035 | XmlNode xn = xmlDoc.SelectSingleNode(“Imageset”); |
| 036 |  |
| 037 | XmlNodeList xnl = xn.ChildNodes; |
| 038 |  |
| 039 | **foreach** (XmlNode xnf **in** xnl) |
| 040 |  |
| 041 | { |
| 042 |  |
| 043 | XmlElement xe = (XmlElement)xnf; |
| 044 |  |
| 045 | *// Debug.Log(“Name=” + xe.GetAttribute(“Name”));* |
| 046 |  |
| 047 | *// Debug.Log(“\t”);* |
| 048 |  |
| 049 | *// Debug.Log(“xpos=” + xe.GetAttribute(“XPos”));* |
| 050 |  |
| 051 | *// Debug.Log(“\t”);* |
| 052 |  |
| 053 | *// Debug.Log(“ypos=” + xe.GetAttribute(“YPos”));* |
| 054 |  |
| 055 | *// Debug.Log(“\t”);* |
| 056 |  |
| 057 | *// Debug.Log(“width=” + xe.GetAttribute(“Width”));* |
| 058 |  |
| 059 | *// Debug.Log(“\t”);* |
| 060 |  |
| 061 | *// Debug.Log(“height=” + xe.GetAttribute(“Height”));* |
| 062 |  |
| 063 | *//Unity3D教程手册：www.unitymanual.com* |
| 064 |  |
| 065 | picName = xe.GetAttribute(“Name”); |
| 066 |  |
| 067 | picPos\_x = Convert.ToInt(xe.GetAttribute(“XPos”)); |
| 068 |  |
| 069 | picPos\_y = Convert.ToInt(xe.GetAttribute(“YPos”)); |
| 070 |  |
| 071 | picWidth = Convert.ToInt(xe.GetAttribute(“Width”)); |
| 072 |  |
| 073 | picHeight = Convert.ToInt(xe.GetAttribute(“Height”)); |
| 074 |  |
| 075 | newTexture = new TextureD(picWidth,picHeight); |
| 076 |  |
| 077 | **for**(**int** m=picPos\_y;m<picPos\_y+picHeight;++m) |
| 078 |  |
| 079 | { |
| 080 |  |
| 081 | **for**(**int** n=picPos\_x;n<picPos\_x+picWidth;++n) |
| 082 |  |
| 083 | { |
| 084 |  |
| 085 | color = resTexture.GetPixel(n,resTexture.height-m); |
| 086 |  |
| 087 | newTexture.SetPixel(n-picPos\_x,picHeight-(m-picPos\_y),color); |
| 088 |  |
| 089 | } |
| 090 |  |
| 091 | } |
| 092 |  |
| 093 | newTexture.Apply(); |
| 094 |  |
| 095 | **byte**[] b = newTexture.EncodeToPNG(); |
| 096 |  |
| 097 | File.WriteAllBytes(“Assets/Resources/**out**/”+picName+“.png”,b); |
| 098 |  |
| 099 | } |
| 100 |  |
| 101 | } |
| 102 |  |
| 103 | } |