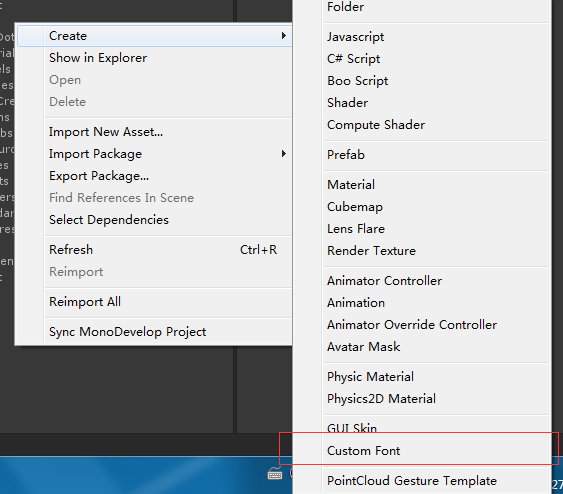
# **UGUI中创建自定义图片字体**

[](http://www.jianshu.com/u/9695b7a6d684)

[重装机霸](http://www.jianshu.com/u/9695b7a6d684) 关注

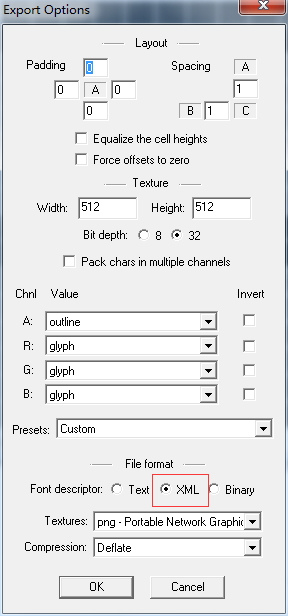
2015.12.08 10:47 字数 529 阅读 3698评论 2喜欢 7

NGUI中，使用自定义字体可以使用font maker，但是UGUI中并没有提供相应工具。  
UGUI的自定义字体为



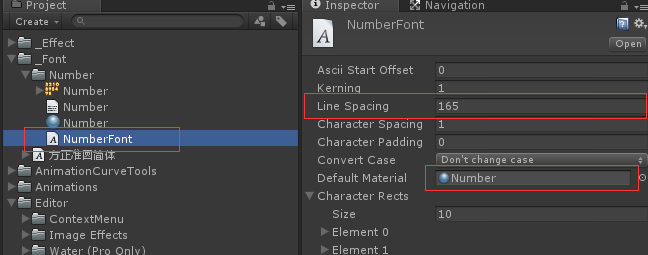
Unity3d自定义字体

并且官方没有提供相对应的字体制作工具。这里提供一个生成custom Font参数的脚本。  
需要配合BMFont使用。  
BMFont导出设置为XML文件

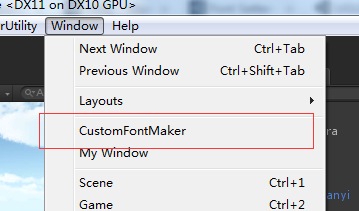


导出文字设置

将生产的.fnt文件和贴图文件拖入项目中，在项目中创建metial，贴图为生成的贴图文件。shader为UI使用的SHADER。  
创建自定义font，将材质绑定到Metial绑定到font中，其中，line spacing为文字的行高，根据实际设置。



打开custom font maker窗口。



将自定义字体以及.font文件拖到指定位置。然后点击创建字体，即可生成对应的字体文件。  
不足之处，没有设置到对应的line spacing。需要创建好后手动设置。

using UnityEngine;

using System.Collections;

using UnityEditor;

using System.Xml;

using System.Text;

using System.IO;

public class CustomFontMaker : EditorWindow

{

private Font font;

private TextAsset xmlText;

[MenuItem("Window/CustomFontMaker")]

static void AddWindow()

{

//创建窗口

Rect wr = new Rect(0, 0, 500, 500);

CustomFontMaker window = (CustomFontMaker)EditorWindow.GetWindowWithRect(typeof(CustomFontMaker), wr, true, "自定义字体");

window.Show();

}

// Use this for initialization

void Start()

{

}

void OnGUI()

{

font = EditorGUILayout.ObjectField("字体", font, typeof(Font), true) as Font;

xmlText = EditorGUILayout.ObjectField("文字XML配置", xmlText, typeof(TextAsset), true) as TextAsset;

if (GUILayout.Button("创建字体", GUILayout.Width(200)))

{

this.CreateFont();

}

}

void CreateFont()

{

XmlDocument \_doc = new XmlDocument();

byte[] \_array = Encoding.ASCII.GetBytes(xmlText.text);

MemoryStream \_stream = new MemoryStream(\_array);

\_doc.Load(\_stream);

XmlNode \_font = \_doc.SelectSingleNode("font");

XmlElement \_common = (XmlElement)\_font.SelectSingleNode("common");

float \_scaleW = float.Parse(\_common.GetAttribute("scaleW"));

float \_scaleH = float.Parse(\_common.GetAttribute("scaleH"));

XmlNode \_chars = \_font.SelectSingleNode("chars");

XmlNodeList \_charsList = \_chars.ChildNodes;

CharacterInfo[] \_infos = new CharacterInfo[\_charsList.Count];

for (int i = 0; i < \_charsList.Count; i++)

{

XmlElement \_element = (XmlElement)\_charsList[i];

CharacterInfo \_characterInfo = new CharacterInfo();

\_characterInfo.index = int.Parse(\_element.GetAttribute("id"));

float \_x = float.Parse(\_element.GetAttribute("x"));

float \_y = float.Parse(\_element.GetAttribute("y"));

int \_width = int.Parse(\_element.GetAttribute("width"));

int \_height = int.Parse(\_element.GetAttribute("height"));

int \_xadvance = int.Parse(\_element.GetAttribute("xadvance"));

\_characterInfo.uv = new Rect(\_x / \_scaleW, 1 - (\_y + \_height) / \_scaleH, \_width / \_scaleW, \_height / \_scaleH);

\_characterInfo.vert = new Rect(0, 0, \_width, -\_height);

\_characterInfo.width = \_xadvance;

Debug.Log(\_characterInfo.uv);

\_infos[i] = \_characterInfo;

}

font.characterInfo = \_infos;

}

}

[Unity3d](http://www.jianshu.com/nb/2524523)

using UnityEngine;

using System.Collections;

using UnityEditor;

using System.Xml;

using System.Text;

using System.IO;

public class CustomFontMaker : EditorWindow

{

private Font font;

private TextAsset xmlText;

[MenuItem("Window/CustomFontMaker")]

static void AddWindow()

{

//创建窗口

Rect wr = new Rect(0, 0, 500, 500);

CustomFontMaker window = (CustomFontMaker)EditorWindow.GetWindowWithRect(typeof(CustomFontMaker), wr, true, "自定义字体");

window.Show();

}

// Use this for initialization

void Start()

{

}

void OnGUI()

{

font = EditorGUILayout.ObjectField("字体", font, typeof(Font), true) as Font;

xmlText = EditorGUILayout.ObjectField("文字XML配置", xmlText, typeof(TextAsset), true) as TextAsset;

if (GUILayout.Button("创建字体", GUILayout.Width(200)))

{

this.CreateFont();

}

}

void CreateFont()

{

XmlDocument \_doc = new XmlDocument();

byte[] \_array = Encoding.ASCII.GetBytes(xmlText.text);

MemoryStream \_stream = new MemoryStream(\_array);

\_doc.Load(\_stream);

XmlNode \_font = \_doc.SelectSingleNode("font");

XmlElement \_common = (XmlElement)\_font.SelectSingleNode("common");

float \_scaleW = float.Parse(\_common.GetAttribute("scaleW"));

float \_scaleH = float.Parse(\_common.GetAttribute("scaleH"));

XmlNode \_chars = \_font.SelectSingleNode("chars");

XmlNodeList \_charsList = \_chars.ChildNodes;

CharacterInfo[] \_infos = new CharacterInfo[\_charsList.Count];

for (int i = 0; i < \_charsList.Count; i++)

{

XmlElement \_element = (XmlElement)\_charsList[i];

CharacterInfo \_characterInfo = new CharacterInfo();

\_characterInfo.index = int.Parse(\_element.GetAttribute("id"));

float \_x = float.Parse(\_element.GetAttribute("x"));

float \_y = float.Parse(\_element.GetAttribute("y"));

int \_width = int.Parse(\_element.GetAttribute("width"));

int \_height = int.Parse(\_element.GetAttribute("height"));

int \_xadvance = int.Parse(\_element.GetAttribute("xadvance"));

\_characterInfo.uv = new Rect(\_x / \_scaleW, 1 - (\_y + \_height) / \_scaleH, \_width / \_scaleW, \_height / \_scaleH);

\_characterInfo.vert = new Rect(0, 0, \_width, -\_height);

\_characterInfo.width = \_xadvance;

Debug.Log(\_characterInfo.uv);

\_infos[i] = \_characterInfo;

}

font.characterInfo = \_infos;

}

}