# Unity3D开发之播放gif图

Unity是不识别Gif格式图的，需要我们使用c#将gif里多帧图转化为Texture2D格式。需要使用System.Drawing.dll.此dll在unity安装目录下就可以找到。由于unity没有gif格式的文件，所以我们无法在面板指定，需要动态加载。所以将gif图放在StreamingAssets文件夹下。以下为源代码：

public class PlayGif : MonoBehaviour {

public UnityEngine.UI.Image Im;

public string gifName = "";

public GameObject[] Ims;

[SerializeField]

private float fps = 5f;

private List<Texture2D> tex2DList = new List<Texture2D>();

private float time;

Bitmap mybitmp;

void Start() {

System.Drawing.Image image = System.Drawing.Image.FromFile(Application.streamingAssetsPath + "/"+gifName+".gif");

tex2DList = MyGif(image);

}

void Update() {

if (tex2DList.Count > 0) {

time += Time.deltaTime;

int index = (int)(time \* fps) % tex2DList.Count;

if (Im != null) {

Im.sprite = Sprite.Create(tex2DList[index], new Rect(0, 0, tex2DList[index].width, tex2DList[index].height), new Vector2(0.5f, 0.5f));

}

if (Ims.Length != 0) {

for (int i = 0; i < Ims.Length; i++)

Ims[i].GetComponent<Renderer>().material.mainTexture = tex2DList[index];

}

}

}

private List<Texture2D> MyGif(System.Drawing.Image image) {

List<Texture2D> tex = new List<Texture2D>();

if (image != null) {

//Debug.Log("图片张数：" + image.FrameDimensionsList.Length);

FrameDimension frame = new FrameDimension(image.FrameDimensionsList[0]);

int framCount = image.GetFrameCount(frame);//获取维度帧数

for (int i = 0; i < framCount; ++i) {

image.SelectActiveFrame(frame, i);

Bitmap framBitmap = new Bitmap(image.Width, image.Height);

using (System.Drawing.Graphics graphic = System.Drawing.Graphics.FromImage(framBitmap)) {

graphic.DrawImage(image, Point.Empty);

}

Texture2D frameTexture2D = new Texture2D(framBitmap.Width, framBitmap.Height, TextureFormat.ARGB32, true);

frameTexture2D.LoadImage(Bitmap2Byte(framBitmap));

tex.Add(frameTexture2D);

}

}

return tex;

}

private byte[] Bitmap2Byte(Bitmap bitmap) {

using (MemoryStream stream = new MemoryStream()) {

// 将bitmap 以png格式保存到流中

bitmap.Save(stream, ImageFormat.Png);

// 创建一个字节数组，长度为流的长度

byte[] data = new byte[stream.Length];

// 重置指针

stream.Seek(0, SeekOrigin.Begin);

// 从流读取字节块存入data中

stream.Read(data, 0, Convert.ToInt32(stream.Length));

return data;

}

}

}

---------------------

原文：https://blog.csdn.net/qq\_33994566/article/details/86534969