纹理效用

基础手册 v1.0

介绍

这个漂亮的软件包为您提供了 25+强大而方便的实用程序来操作 Texture2D 资产。它允许在运行时更改像素级别的纹理，以实现许多复杂和有用的效果。这极大地扩展了 Unity 更改纹理的内置功能，带来了诸如:

- 应用蒙版和布尔操作
- 旋转，缩放，镜像，扩展，裁剪和自动裁剪
- 改变对比度，亮度，色调，饱和度
- 上色、灰度、负色和离开色彩调和
- 反转透明，使颜色透明
- 笔画等

如何使用

用法非常简单直接:

1. 准备你想要使用的纹理资源。这可以手动完成，也可以使用纹理实用性等函数。
   PrepareAsset(_source: Texture2D);-*用于现有纹理*的纹理实用性。CreateTexture (_width: int， _height: int， _color: Color);-*创建新的纹理*

   最重要的是——纹理应该是可读的!

2. 从*纹理实用性名称空间调用所需*的函数。例如:纹理实用性。(你的 sourcetexture);纹理实用性。旋转(你的 sourcetexture, desire 旋转角);纹理实用性。autocrop transparent(你的 sourcetexture);

   更多关于函数和参数的信息请在下一页查看。

提示和提示

1. 一些函数有不同的版本-对于 *Texture2D* 和 *Color[]*源。如果你需要经常调用函数，建议使用 *Color[]* 版本(这将节省内存和 CPU，特别是如果你将相同的像素数组传递给几个函数)。

2. 在纹理导入设置中最好有:*textureFormat = RGBA32, isReadable = true;mipmapEnabled = false;alphaitransparency = true;*

3. 你可以改变 Texture2D 资产本身，甚至在 unity 中编辑图像。

   要将新图像保存为资产，请使用 TextureUtility 函数。SaveAsAsset(_source: Texture2D， _path: String)

4. 注意不要覆盖源纹理。建议从一开始就创建/使用 copy。

5. 为了您的舒适，一些函数(ApplyBooleanOperation 作为实例)可以隐式地改变结果纹理宽度/高度。请考虑到这一点。

6. 一些使用透明度操作的函数(如 ApplyMask, ApplyBooleanOperation, AutoCropTransparency)使用二进制值(1 或 0)，因此为了获得正确的结果，透明度 alpha 应该恰好为 0(即半透明像素将被视为不透明)

*如果您有任何问题/问题-请不要犹豫通过 **AllebiGames@gmail.com** 联系我*

功能和语法

用于更方便设置的枚举:

布尔运算:并、交、减法

Union - *result Texture 将包含两个纹理的所有非透明像素*Intersection - *result Texture 将只包含纹理的重叠非透明像素*

*减法-结果纹理将只包含不重叠的非透明像素***对齐**:TopLeft, TopRight, TopCenter, BottomLeft, BottomRight, BottomCenter。**BlendMode**:法线，加法，减法，乘法，细分，MaskAlpha

资产准备功能:

*准备通过TextureUtility 操作纹理(修改导入器设置)***PrepareAsset** (_source: Texture2D): Texture2D

将**纹理作为资源保存到 Assets 文件夹 SaveAsAsset** (_source: Texture2D， _path:String)中的路径中。

创建新的Texture2D/PixelArray 填充自定义颜色

CreateTexture (_width: int， _height: int， _color: Color): Texture2D
CreatePixelArray (_width: int， _height: int， _color: Color): Color[]

操作颜色等的函数:

Clear **(_source: Texture2D ， _color**: color): Texture2D
Clear (_source: color []， _color: color): color []

倒置 Texture2D/PixelArray 透明度 **InvertTransparency** (_source: Texture2D): Texture2D **InvertTransparency** (_source: Color[]): Color[]

使Texture2D/PixelArray 的自定义颜色完全透明(Alpha-key)
MakeColorTransparent (_source: Texture2D， _color: color): Texture2D
MakeColorTransparent (_source: color []， _color: color): color []

*更改 Texture2D/PixelArray HSB(允许调整色相，饱和度和亮度)*更改 **HSB**(_source: Texture2D， _hue: float， _saturation: float， _brightness: float): Texture2D **更改 HSB**(_source: Color[]， _hue: float， _saturation: float， _brightness:float): Color[]

改变Texture2D/PixelArray 的对比度

对比(_source: Texture2D， _contrast: float): Texture2D **对比** (_source: Color[]， _contrast: float): Color[]

在Texture2D/PixelArray 中保留仅在自定义diapson 内具有颜色的像素(所有其他像素变为灰度)

ColorDiapason (_source: Texture2D， _colorStart: Color， _colorEnd: Color): Texture2D
ColorDiapason (_source: Color[]， _colorStart: Color， _colorEnd: Color): Color[]

Colorize **(_source: Texture2D， _color: color， _intensity**: float): Texture2D
Colorize (_source: color []， _color: color， _intensity: float): color []

将Texture2D/PixelArray 转换为灰度

灰度(_source: Texture2D): Texture2D

灰度(_source: Color[]): Color[]

将 Texture2D/PixelArray 转换为自身的负值

negative (_source: Texture2D): Texture2D

negative (_source: Color[]): Color[]

操作纹理几何等的函数:

水平翻转 Texture2D

flihorizontal (**_source**: Texture2D): Texture2D

垂直翻转 Texture2D

flipvertical (_source: Texture2D): Texture2D

改变 Texture2D 画布的宽度/高度通过扩展在指定的方向(函数不缩放图像本身)

扩展(_source: Texture2D， _newWidth: int， _newHeight: int， _sourceAlignment: Alignment): Texture2D

通过在指定矩形中提取 Texture2D 块来裁剪它

裁剪(_source: Texture2D， _rect: Rect):纹理

自动裁剪图像周围的透明像素

AutoCropTransparency (_source: Texture2D): Texture2D

自动裁剪像素(自定义颜色)周围的图像 **AutoCropColor**
(_source: Texture2D， _color: color): Texture2D

应用透明(Alpha-chanel)蒙版到 Texture2D **应用 ApplyMask**
(_source: Texture2D， _mask: Texture2D): Texture2D

通过 apply 布尔操作(基于 Alpha-channel)来合并 2 个纹理。Result-texture 的大小会在需要时扩展。

ApplyBooleanOperation (_operationType: BooleanOperation， _base: Texture2D， _addition: Texture2D，
_additionOffset: Vector2): Texture2D

*旋转 Texture2D 自定义角度***旋转**(_source: Texture2D，
_angle: float): Texture2D

用新的宽度/高度缩放 Texture2D

Scale (_source: Texture2D， _targetWidth: int， _targetHeight: int): Texture2D

*描边(为边缘像素上色)Texture2D 通过指定厚度和 blendMode 的颜色***描**边(_source: Texture2D，
_thickness: int， _color: color， _blendMode: blendMode): Texture2D

计算/转换颜色的附属函数:

*在 PixelsArray***GetAverageColor**(_pixels:
color[])中获取平均颜色:color

使用 blendmode 中的一种混合 2 种颜色

BlendColors (_color1: Color， _color2: Color， _blendMode: blendMode): Color

Color 转换为 HSBA (Vector4)
ColorToHSBA (_color: Color): Vector4

转换 HSBA(Vector4)为颜色
HSBAtoColor (HSBA: Vector4):颜色