

孤独の巡礼

昵称：太乙_真人
园龄：12年2个月
粉丝：28
关注：0
[+加关注](#)

< 2025年6月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

搜索

找找看

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

- 随笔档案
- 2025年3月(1)

2024年6月(1)

2024年4月(1)

2023年11月(1)

2023年10月(1)

2023年8月(3)

2023年7月(2)

2023年5月(1)

2023年4月(3)

2023年3月(5)

2023年2月(4)

2023年1月(21)

2022年11月(1)

2022年9月(1)

2022年6月(1)

更多

- 阅读排行榜
1. Unity3d之将terrain转化成mesh(15418)

2. c#之如何转换文本文件编码格式为utf-8(12123)

3. Unity如何退出游戏(11155)

4. 四元数小总结(9945)

5. Unity shader之ColorMask(8333)

- 评论排行榜
1. Unity 之事件系统(6)

2. Unity3d之将terrain转化成mesh(4)

3. Unity性能优化之特效合并(3)

4. Unity之如何从fbx提取Animation clip文件(3)

5. unity3d 多人寻路堵塞堆叠问题(3)

- 推荐排行榜
-

CG标准函数库

(1) 数学函数

函数	功能描述
<code>abs(x)</code>	返回输入参数的绝对值
<code>acos(x)</code>	反余切函数，输入参数范围为[-1,1]， 返回[0,π]区间的角度值
<code>all(x)</code>	如果输入参数均不为0，则返回ture； 否则返回flase。&&运算
<code>any(x)</code>	输入参数只要有其中一个不为0，则返回true。
<code>asin(x)</code>	反正弦函数,输入参数取值区间为[-1,1]， 返回角度值范围为, [-π2,π2]
<code>atan(x)</code>	反正切函数， 返回角度值范围为[-π2,π2]
<code>atan2(y,x)</code>	计算y/x的反正切值。实际上和atan(x)函数功能完全一样，至少输入参数不同。atan(x) = atan2(x, float(1))。
<code>ceil(x)</code>	对输入参数向上取整。例如： ceil(float(1.3))， 其返回值为2.0
<code>clamp(x,a,b)</code>	如果x值小于a，则返回a； 如果x值大于b，返回b； 否则，返回x。
<code>cos(x)</code>	返回弧度x的余弦值。返回值范围为[-1,1]
<code>cosh(x)</code>	双曲余弦（hyperbolic cosine）函数，计算x的双曲余弦值。
<code>cross(A,B)</code>	返回两个三元向量的叉积(cross product)。注意，输入参数必须是三元向量！
<code>degrees(x)</code>	输入参数为弧度值(radians)，函数将其转换为角度值(degrees)
<code>determinant(m)</code>	计算矩阵的行列式因子。
<code>dot(A,B)</code>	返回A和B的点积(dot product)。参数A和B可以是标量，也可以是向量（输入参数方面，点积和叉积函数有很大不同）。
<code>exp(x)</code>	计算ex的值，e=2.71828182845904523536
<code>exp2(x)</code>	计算2x的值
<code>floor(x)</code>	对输入参数向下取整。例如floor(float(1.3))返回的值为1.0；但是floor(float(-1.3))返回的值为-2.0。该函数与ceil(x)函数相对应。
<code>fmod(x,y)</code>	返回x/y的余数。如果y为0，结果不可预料。
<code>frac(x)</code>	返回标量或矢量的小数
<code>frexp(x, out i)</code>	将浮点数x分解为尾数和指数，即x=m*2 ⁱ ， 返回m，并将指数存入i中；如果x为0，则尾数和指数都返回0
<code>isfinite(x)</code>	判断标量或者向量中的每个数据是否是有限数，如果是返回true；否则返回false;

1. c#如何判断字符串是否含中文(2)
2. Unity3d之将terrain转化成mesh(2)
3. Jenkins+Unity自动化打包之如何跑通第一个shell脚本打印Hello world(1)
4. Unity shader学习之简单的水效果(1)
5. Unity 之事件系统(1)

最新评论

1. Re:Unity3d 开发之 ulua 坑的总结【Unity3D】Unity3D技术栈
1. Re:Unity3d之将terrain转化成mesh @马马呼呼 var t = terrainObj.transfor m.parent.Find(transName); 可能是Terrain没有parent导致的，给Terrain一个父节点试试...
--little_fat_sheep
2. Re:Unity3d之将terrain转化成mesh 为啥代码复制进去之后 报错了 报了个这个 NullReferenceException: Object reference not set to an instance of an object Te...
--孤独的巡礼
3. Re:Unity3d之将terrain转化成mesh 为啥代码复制进去之后 报错了 报了个这个 NullReferenceException: Object reference not set to an instance of an object Te...
--马马呼呼
4. Re:Unity之如何从fbx提取Animation clip文件
大佬，AnimationClip直接在Fbx里压缩和去除Scale不是也可以吗，为啥要把AnimationClip给复制出一份呢？
--Li_sj
5. Re:Unity 之 Addressable Asset System 之工具创建group
你好。你可以给我一个下载这个项目download link 或 GitHub link 吗？
--MikakuTosu

函数	功能描述																																																																																																		
isinf(x)	判断标量或者向量中的每个数据是否是无限，如果是返回true；否则返回false；																																																																																																		
isnan(x)	判断标量或者向量中的每个数据是否是非数据(not-a-number NaN)，如果是返回true；否则返回false；																																																																																																		
ldexp(x, n)	计算x*2n的值																																																																																																		
lerp(a, b, f)	计算(1-f)*a+b*f或者a+f*(b-a)的值。即在下限 a 和上限 b 之间进行插值， f 表示权值。注意，如果 a 和 b 是向量，则权值 f 必须是标量或者等长的向量。																																																																																																		
lit(NdotL, NdotH, m)	N 表示法向量； L 表示入射光向量； H 表示半角向量； m 表示高光系数。 函数计算环境光、散射光、镜面光的贡献，返回的4元向量。 X 位表示环境光的贡献，总是1.0； Y 位代表散射光的贡献，如果 N·L<0，则为0；否则为N·L Z 位代表镜面光的贡献，如果N·L<0 或者N·H<0，则位0；否则为(N·L)m； W 位始终位1.0																																																																																																		
log(x)	计算ln(x)的值， x 必须大于 0																																																																																																		
log2(x)	计算log(x)2的值， x 必须大于 0																																																																																																		
log10(x)	计算log(x)10的值， x 必须大于 0																																																																																																		
max(a, b)	比较两个标量或等长向量元素，返回最大值。																																																																																																		
min(a,b)	比较两个标量或等长向量元素，返回最小值。																																																																																																		
modf(x, out ip)	把 x 分解成整数和分数两部分，每部分都和 x 有着相同的符号，整数部分被保存在 ip 中，分数部分由函数返回																																																																																																		
mul(M, N)	矩阵 M 和矩阵 N 的积，计算方法如下 mul(M,N)= <table><tr><td><table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table></td><td><table><tr><td>N</td><td>1</td><td>1</td><td>N</td><td>1</td><td>2</td><td>N</td><td>1</td><td>3</td><td>N</td><td>1</td><td>4</td><td>N</td><td>2</td><td>1</td><td>N</td><td>2</td><td>2</td><td>N</td><td>2</td><td>3</td><td>N</td><td>2</td><td>4</td><td>N</td><td>3</td><td>1</td><td>N</td><td>3</td><td>2</td><td>N</td><td>3</td><td>3</td><td>N</td><td>3</td><td>4</td><td>N</td><td>4</td><td>1</td><td>N</td><td>4</td><td>2</td><td>N</td><td>4</td><td>3</td><td>N</td><td>4</td><td>4</td></tr></table></td></tr></table>	<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4	<table><tr><td>N</td><td>1</td><td>1</td><td>N</td><td>1</td><td>2</td><td>N</td><td>1</td><td>3</td><td>N</td><td>1</td><td>4</td><td>N</td><td>2</td><td>1</td><td>N</td><td>2</td><td>2</td><td>N</td><td>2</td><td>3</td><td>N</td><td>2</td><td>4</td><td>N</td><td>3</td><td>1</td><td>N</td><td>3</td><td>2</td><td>N</td><td>3</td><td>3</td><td>N</td><td>3</td><td>4</td><td>N</td><td>4</td><td>1</td><td>N</td><td>4</td><td>2</td><td>N</td><td>4</td><td>3</td><td>N</td><td>4</td><td>4</td></tr></table>	N	1	1	N	1	2	N	1	3	N	1	4	N	2	1	N	2	2	N	2	3	N	2	4	N	3	1	N	3	2	N	3	3	N	3	4	N	4	1	N	4	2	N	4	3	N	4	4
<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4	<table><tr><td>N</td><td>1</td><td>1</td><td>N</td><td>1</td><td>2</td><td>N</td><td>1</td><td>3</td><td>N</td><td>1</td><td>4</td><td>N</td><td>2</td><td>1</td><td>N</td><td>2</td><td>2</td><td>N</td><td>2</td><td>3</td><td>N</td><td>2</td><td>4</td><td>N</td><td>3</td><td>1</td><td>N</td><td>3</td><td>2</td><td>N</td><td>3</td><td>3</td><td>N</td><td>3</td><td>4</td><td>N</td><td>4</td><td>1</td><td>N</td><td>4</td><td>2</td><td>N</td><td>4</td><td>3</td><td>N</td><td>4</td><td>4</td></tr></table>	N	1	1	N	1	2	N	1	3	N	1	4	N	2	1	N	2	2	N	2	3	N	2	4	N	3	1	N	3	2	N	3	3	N	3	4	N	4	1	N	4	2	N	4	3	N	4	4		
M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4																																																				
N	1	1	N	1	2	N	1	3	N	1	4	N	2	1	N	2	2	N	2	3	N	2	4	N	3	1	N	3	2	N	3	3	N	3	4	N	4	1	N	4	2	N	4	3	N	4	4																																																				
mul(M, v)	矩阵 M 和列向量 v 的积，公式如下 mul(M,v)= <table><tr><td><table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table></td><td><table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table></td></tr></table>	<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4	<table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table>	v	1	v	2	v	3	v	4																																								
<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4	<table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table>	v	1	v	2	v	3	v	4																																										
M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4																																																				
v	1	v	2	v	3	v	4																																																																																												
mul(v, M)	行向量 v 和矩阵 M 的积，公式如下 mul(v,M)= <table><tr><td><table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table></td><td><table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table></td></tr></table>	<table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table>	v	1	v	2	v	3	v	4	<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4																																								
<table><tr><td>v</td><td>1</td><td>v</td><td>2</td><td>v</td><td>3</td><td>v</td><td>4</td></tr></table>	v	1	v	2	v	3	v	4	<table><tr><td>M</td><td>1</td><td>1</td><td>M</td><td>1</td><td>2</td><td>M</td><td>1</td><td>3</td><td>M</td><td>1</td><td>4</td><td>M</td><td>2</td><td>1</td><td>M</td><td>2</td><td>2</td><td>M</td><td>2</td><td>3</td><td>M</td><td>2</td><td>4</td><td>M</td><td>3</td><td>1</td><td>M</td><td>3</td><td>2</td><td>M</td><td>3</td><td>3</td><td>M</td><td>3</td><td>4</td><td>M</td><td>4</td><td>1</td><td>M</td><td>4</td><td>2</td><td>M</td><td>4</td><td>3</td><td>M</td><td>4</td><td>4</td></tr></table>	M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4																																										
v	1	v	2	v	3	v	4																																																																																												
M	1	1	M	1	2	M	1	3	M	1	4	M	2	1	M	2	2	M	2	3	M	2	4	M	3	1	M	3	2	M	3	3	M	3	4	M	4	1	M	4	2	M	4	3	M	4	4																																																				
noise(x)	根据它的参数类型，这个函数可以是一元、二元或三元噪音函数。返回的值在 0 和 1 之间，并且通常与给定的输入值一样																																																																																																		
pow(x, y)	xy																																																																																																		
radians(x)	函数将角度值转换为弧度值																																																																																																		
round(x)	返回四舍五入值。																																																																																																		
rsqrt(x)	x 的平方根的倒数， x 必须大于 0																																																																																																		
saturate(x)	把 x 限制到 [0,1] 之间																																																																																																		
sign(x)	如果x>0则返回 1 ；否则返回 0																																																																																																		
sin(x)	输入参数为弧度，计算正弦值，返回值范围 为 [-1,1]																																																																																																		

函数	功能描述
<code>sincos(float x, out s, out c)</code>	该函数是同时计算x的sin值和cos值，其中s= $\sin(x)$ ，c= $\cos(x)$ 。该函数用于“同时需要计算sin值和cos值的情况”，比分别运算要快很多！
<code>sinh(x)</code>	计算x的双曲正弦
<code>smoothstep(min, max, x)</code>	值x位于min、max区间中。如果x=min，返回0；如果x=max，返回1；如果x在两者之间，按照下列公式返回数据： $-2*(x-minmax-min)^3+3*(x-minmax-min)^2$
<code>step(a, x)</code>	如果x<a，返回0；否则，返回1
<code>sqrt(x)</code>	求x的平方根， $x\sqrt{}$ ，x必须大于0
<code>tan(x)</code>	计算x正切值
<code>tanh(x)</code>	计算x的双曲线切线
<code>transpose(M)</code>	矩阵M的转置矩阵 如果M是一个AxB矩阵，M的转置是一个BxA矩阵，它的第一列是M的第一行，第二列是M的第二行，第三列是M的第三行，等等

(2) 几何函数

函数	功能描述
<code>distance(pt1, pt2)</code>	两点之间的欧几里德距离 (Euclidean distance)
<code>faceforward(N,I, Ng)</code>	如果 $N_g \cdot I < 0$ ，返回N；否则返回-N。
<code>length(v)</code>	返回一个向量的模，即 $\sqrt{\text{dot}(v,v)}$
<code>normalize(v)</code>	返回v向量的单位向量
<code>reflect(I, N)</code>	根据入射光线方向I和表面法向量N计算反射向量，仅对三元向量有效
<code>refract(I,N,eta)</code>	根据入射光线方向I，表面法向量N和折射相对系数eta,计算折射向量。如果对给定的eta,I和N之间的角度太大，返回(0,0,0)。 只对三元向量有效

(3) 纹理映射函数

函数	功能描述
<code>tex1D(sampler1D tex, float s)</code>	一维纹理查询
<code>tex1D(sampler1D tex, float s, float dsdx, float dsdy)</code>	使用导数值 (derivatives) 查询一维纹理
<code>Tex1D(sampler1D tex, float2 sz)</code>	一维纹理查询，并进行深度值比较
<code>Tex1D(sampler1D tex, float2 sz, float dsdx, float dsdy)</code>	使用导数值 (derivatives) 查询一维纹理， 并进行深度值比较
<code>Tex1Dproj(sampler1D tex, float2 sq)</code>	一维投影纹理查询

函数	功能描述
Tex1Dproj(sampler1D tex, float3 szq)	一维投影纹理查询，并比较深度值
Tex2D(sampler2D tex, float2 s)	二维纹理查询
Tex2D(sampler2D tex, float2 s, float2 dsdx, float2 dsdy)	使用导数值（derivatives）查询二维纹理
Tex2D(sampler2D tex, float3 sz)	二维纹理查询，并进行深度值比较
Tex2D(sampler2D tex, float3 sz, float2 dsdx, float2 dsdy)	使用导数值（derivatives）查询二维纹理，并进行深度值比较
Tex2Dproj(sampler2D tex, float3 sq)	二维投影纹理查询
Tex2Dproj(sampler2D tex, float4 szq)	二维投影纹理查询，并进行深度值比较
texRECT(samplerRECT tex, float2 s)	二维非投影矩形纹理查询（OpenGL独有）
texRECT (samplerRECT tex, float3 sz, float2 dsdx, float2 dsdy)	二维非投影使用导数的矩形纹理查询（OpenGL独有）
texRECT (samplerRECT tex, float3 sz)	二维非投影深度比较矩形纹理查询（OpenGL独有）
texRECT (samplerRECT tex, float3 sz, float2 dsdx, float2 dsdy)	二维非投影深度比较并使用导数的矩形纹理查询（OpenGL独有）
texRECT proj(samplerRECT tex, float3 sq)	二维投影矩形纹理查询（OpenGL独有）
texRECT proj(samplerRECT tex, float3 szq)	二维投影矩形纹理深度比较查询（OpenGL独有）
Tex3D(sampler3D tex, float s)	三维纹理查询
Tex3D(sampler3D tex, float3 s, float3 dsdx, float3 dsdy)	结合导数值（derivatives）查询三维纹理
Tex3Dproj(sampler3D tex, float4 szq)	查询三维投影纹理，并进行深度值比较
texCUBE(samplerCUBE tex, float3 s)	查询立方体纹理
texCUBE (samplerCUBE tex, float3 s, float3 dsdx, float3 dsdy)	结合导数值（derivatives）查询立方体纹理
texCUBEproj (samplerCUBE tex, float4 sq)	查询投影立方体纹理

在这个表中，每个函数第二个参数的名字指明了在执行纹理查询的时候，它的值如果是被使用的：

- **s**表示这是一个一元、二元或三元纹理坐标。
- **z**表示这是一个用来进行阴影贴图查找的深度比较值。
- **q**表示这是一个透视值，在进行纹理查找之前，它被用来除以纹理坐标（**s**）。

当你使用的纹理函数允许你指定一个深度比较值的时候，与之相关联的纹理单元必须被设置成深度比较纹理。否则，深度比较实际上不会被执行。

(4) 偏导函数

函数	功能描述
ddx(a)	近似 a 关于屏幕空间 x 轴的偏导数

函数	功能描述
ddy(a)	近似a关于屏幕空间y轴的偏导数

(5) 调试函数

函数	功能描述
void debug(float4 x)	如果在编译时设置了DEBUG，片段着色器中调用该函数可以将值x作为COLOR语义的最终输出；否则该函数什么也不做。

好文要顶

关注我

收藏该文

微信分享



太乙_真人

粉丝 - 28 关注 - 0

00

+加关注

升级成为会员

« 上一篇：[Unity shader学习之Alpha Blend](#)
» 下一篇：[Unity shader学习之Forward Rendering Path](#)

posted @ 2017-07-17 15:01 太乙_真人 阅读(379) 评论(0) 收藏 举报

刷新页面 返回顶部

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

- 【推荐】100%开源！大型工业跨平台软件C++源码提供，建模，组态！
- 【推荐】博客园的心动：当一群程序员决定开源共建一个真诚相亲平台
- 【推荐】国内首个AI IDE，深度理解中文开发场景，立即下载体验Trae
- 【推荐】Flutter适配HarmonyOS 5知识地图，实战解析+高频避坑指南
- 【推荐】凌霞软件回馈社区，携手博客园推出1Panel与Halo联合终身会员
- 【推荐】轻量又高性能的 SSH 工具 IShell：AI 加持，快人一步



- 编辑推荐：
- 智能桌面机器人：使用 .NET 为树莓派开发 Wifi 配网功能
 - C# 模式匹配全解：原理、用法与易错点
 - 记一次SSD性能瓶颈排查之路——寿命与性能之间的取舍
 - 理解 .NET 结构体字段的内存布局
 - .NET 9中的异常处理性能提升分析：为什么过去慢，未来快
- 阅读排行：
- 我的开源十年
 - 【故障公告】6月9日 17:24~17:34 再次遭遇攻击（晚上遭遇更疯狂的攻击）
 - .net入行第4年
 - （包含5w字提示词开源）手把手教你搭建开源提示词优化平台
 - 3 个超火的开源项目「GitHub 热点速览」

