

导航

博客园
首 页
新随笔
联 系
订 阅 
管 理

<	2012年8月						>
日	一	二	三	四	五	六	
29	30	31	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	31	1	
2	3	4	5	6	7	8	

公告

昵称：zdd
园龄：7年
荣誉：推荐博客
粉丝：677
关注：20
+加关注

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

Android(11)
C/C++(19)
D3D10
D3D11(1)
D3D9(4)
Demo(3)
Direct2D(14)
DirectInput(1)
DirectWrite(3)
DirectX(58)
DXGI
HLSL
IOS(2)
Math(16)
OpenGL(1)

几何变换详解

在三维图形学中，几何变换大致分为三种，平移变换（Translation），缩放变换（Scaling），旋转变换（Rotation）。以下讨论皆针对DirectX，所以使用左手坐标系。

平移变换

将三维空间中的一个点[x, y, z, 1]移动到另外一个点[x', y', z', 1]，三个坐标轴的移动分量分别为dx=Tx, dy=Ty, dz=Tz，即

$$x' = x + Tx$$

$$y' = y + Ty$$

$$z' = z + Tz$$

平移变换的矩阵如下。

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{bmatrix}$$

缩放变换

将模型放大或者缩小，本质也是对模型上每个顶点进行放大和缩小（顶点坐标值变大或变小），假设变换前的点是[x, y, z, 1]，变换后的点是[x', y', z', 1]，那么

$$x' = x * Sx$$

$$y' = y * Sy$$

$$z' = z * Sz$$

缩放变换的矩阵如下。

Perl(1)
PS(1)
Shader(2)
Swift
Unity3D
Vim
XAudio2
编程语言(9)
代码片段(19)
数据结构与算法(26)
图形学(10)
移动开发(10)
杂(24)

随笔档案

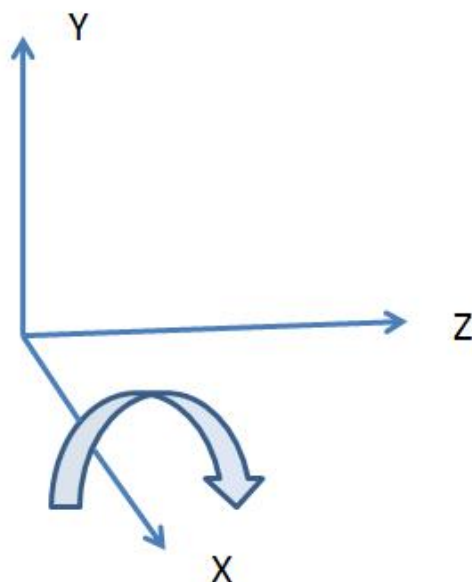
2016年1月 (4)
2015年12月 (1)
2015年11月 (1)
2015年10月 (2)
2015年8月 (5)
2015年7月 (1)
2014年11月 (2)
2014年6月 (2)
2014年5月 (1)
2013年9月 (1)
2013年7月 (1)
2013年6月 (1)
2013年4月 (2)
2013年3月 (3)
2013年2月 (3)
2012年11月 (2)
2012年10月 (2)
2012年9月 (5)
2012年8月 (6)
2012年7月 (5)
2012年6月 (2)
2012年5月 (2)
2012年4月 (1)
2011年9月 (2)
2011年7月 (3)
2011年6月 (2)
2011年5月 (4)
2011年4月 (3)
2011年3月 (3)
2011年2月 (4)
2010年12月 (3)
2010年11月 (3)
2010年9月 (1)
2010年8月 (11)
2010年7月 (17)
2010年6月 (9)
2010年5月 (12)
2010年4月 (2)
2010年3月 (16)
2010年2月 (3)
2010年1月 (2)
2009年12月 (2)
2009年11月 (9)

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

旋转变换

这是三种变换中最复杂的变换，这里只讨论最简单的情况，绕坐标轴旋转，关于绕任意轴旋转，在后续的随笔中介绍。

绕X轴旋转



绕X轴旋转时，顶点的x坐标不发生变化，y坐标和z坐标绕X轴旋转 θ 度，旋转的正方向为顺时针方向（沿着旋转轴负方向向原点看）。 $[x, y, z, 1]$ 表示变换前的点， $[x', y', z', 1]$ 表示变换后的点。变换矩阵如下。

关于旋转的正方向，OpenGL与多数图形学书籍规定旋转正方向为逆时针方向（沿着坐标轴负方向向原点看），比如Computer Graphics C Version, p409。

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2009年10月 (11)
2009年9月 (3)
2009年8月 (1)
2009年7月 (4)
2009年6月 (10)
2009年5月 (2)

Game Engine

DX11 Tutorials
Irrlicht
MathWords
Ogre

OpenGL

缤纷世界

3D Controls
3D Fractal
chaos files
ChaosinChinese
Cloth Simulation
DirectX Developer Center
DirectX document online
DX tutorials
EuclideanSpace
Fractal Video
geometrictools
Google C++ Style
HardCode
HyperGraph
In Framez
LatexEditor
Mame
Mandelbrot Set
OpenGL official page
OpenGL Tutorials
pouet
Ray tracing
Rthdribl
TechInterview
toymaker
W3SCHOOL
XNA/DirectX Forum
X-Zone
云风的Blog

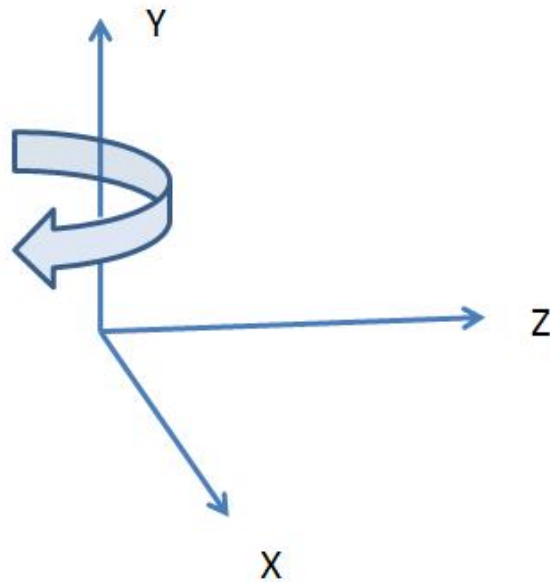
其他

Aogo汇编小站
Channel 9
Code all in one
Emath
Math Circle
Math Game
Microsoft At Home
Microsoft At Work
Windows forum
中国DOS联盟

友情链接

LittleStart
WW老弟

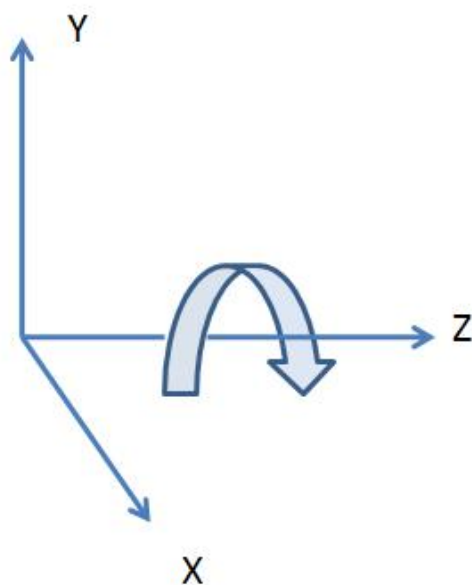
绕Y轴旋转



绕Y轴旋转时，顶点的y坐标不发生变化，x坐标和z坐标绕Y轴旋转 θ 度。 $[x, y, z, 1]$ 表示变换前的点， $[x', y', z', 1]$ 表示变换后的点。变换矩阵如下。

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

绕Z轴旋转



积分与排名

积分 - 199994

排名 - 780

最新评论

1. Re:C++ 初始化列表
31楼说的对。
我就觉得这里有问题，自己也试了，确实不行。看评论找了半天，看到有一样的了。
楼主抓紧重新编辑下啊。

--peiruibo

2. Re:Quaternion和Rotation Matrix的相互转换
我是想用手机开发个东西,这两个属性在性能上有什么区别啊

--普通男孩

3. Re:Quaternion和Rotation Matrix的相互转换
您好我是个小小白,最近看一个大神的代码里面写的是这样的if (this.useQuaternions) { deviceQuat = createQuaternion(alpha, bet.....

--普通男孩

4. Re:算法-求二进制数中1的个数
@zdd我的不是最高的，我比你的快速法，少算了个减法...

--lijunwyf

5. Re:算法-求二进制数中1的个数
@zdd你说的是对的，你有个算法跟我相同。用指令是最快的...

--lijunwyf

阅读排行榜

1. 点到平面的距离公式(41741)
2. 算法-求二进制数中1的个数(41372)
3. C++ 初始化列表(39921)
4. 字符串面试题（一）字符串逆序(38494)
5. 关于数组的几道面试题(36279)

评论排行榜

1. 程序员，请昂起你高贵的头！（364）
2. 关于数组的几道面试题(96)
3. 几何变换详解(76)

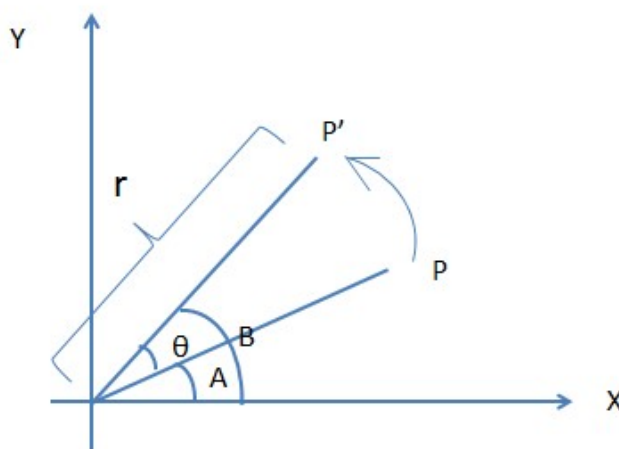
绕Z轴旋转时，顶点的z坐标不发生变化，x坐标和y坐标绕Z轴旋转 θ 度。 $[x, y, z, 1]$ 表示变换前的点， $[x', y', z', 1]$ 表示变换后的点。变换矩阵如下。

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

绕坐标轴旋转的矩阵推导

上面三个旋转矩阵是如何得来的呢？我们推导一下，首先看一下二维的情况，再扩展到三维即可。实际上上面三种绕坐标轴旋转的情况属于特殊的二维旋转，比如绕Z轴旋转，相当于在与XOY平面上绕原点做二维旋转。

假设点P(x, y)是平面直角坐标系内一点，其到原点的距离为r，其与X轴的夹角为A，现将点P绕原点旋转 θ 度，得到点P'(x', y')，P'与X轴的夹角为B，则A = B - θ 。（注意，在二维坐标中，逆时针旋转时角度为正，顺时针旋转时角度为负，下图中由P旋转到P'，角度为 θ ，若是由P'转到P，则角度为 $-\theta$ ）。



$$r = (x, y) = (r \cos A + r \sin A)$$

$$p' = (x', y') = (r \cos B, r \sin B) = (r \cos(A + \theta), r \sin(A + \theta))$$

$$r \cos(A + \theta) = r \cos A \cos \theta - r \sin A \sin \theta = x \cos \theta - y \sin \theta$$

$$r \sin(A + \theta) = r \sin A \cos \theta + r \cos A \sin \theta = y \cos \theta + x \sin \theta$$

于是可得下面的转换方程

4. 判断点是否在三角形内(62)
5. 使用DirectX截屏(55)

推荐排行榜

1. 程序员，请昂起你高贵的头！(174)
2. 算法-求二进制数中1的个数(28)
3. C++ 初始化列表(25)
4. 字符串面试题（一）字符串逆序(20)
5. 关于数组的几道面试题(17)

$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = y\cos\theta + x\sin\theta \\ z' = z \end{cases} \quad (\text{式一})$$

写成矩阵的形式就是

$$(x', y', z') = (x, y, z) \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

求得旋转矩阵为

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

由于这里使用齐次坐标，所以还需加上一维，最终变成如下形式

绕Z轴旋转矩阵

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

和前面给出的绕Z轴旋转矩阵完全吻合。

对于绕X轴旋转的情况，我们只需将式一中的x用y替换，y用z替换，z用x替换即可。替换后得到

$$\begin{cases} y' = y\cos\theta - z\sin\theta \\ z' = z\cos\theta + y\sin\theta \\ x' = x \end{cases} \quad (\text{式二})$$

对应的旋转矩阵为

绕X轴旋转矩阵

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

对于绕Y轴旋转的情况，只需对式二做一次同样的替换即可，的到的变换方程为

$$\begin{cases} z' = z\cos\theta - x\sin\theta \\ x' = x\cos\theta + z\sin\theta \\ y' = y \end{cases}$$

对应的变换矩阵为

绕Y轴旋转矩阵

$$\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

逆矩阵

平移变换矩阵的逆矩阵与原来的平移量相同，但是方向相反。

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -T_x & -T_y & -T_z & 1 \end{bmatrix} = I$$

旋转变换矩阵的逆矩阵与原来的旋转轴相同但是角度相反。

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

$$\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

缩放变换的逆矩阵正好和原来的效果相反，如果原来是放大，则逆矩阵是缩小，如果原来是缩小，则逆矩阵是放大。

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{S_x} & 0 & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 & 0 \\ 0 & 0 & \frac{1}{S_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

== Happy Coding ==

出处：<http://www.cnblogs.com/graphics/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [DirectX](#)

好文要顶

关注我

收藏该文



zdd

关注 - 20

粉丝 - 677

荣誉：推荐博客

[+加关注](#)

4

0

(请您对文章做出评价)

« 上一篇：[矩阵-DirectX与OpenGL的不同](#)

» 下一篇：[绕任意轴旋转](#)

posted on 2012-08-08 09:23 zdd 阅读(11931) 评论(76)

[编辑](#) [收藏](#)

[< Prev](#)

[1](#)

[2](#)

评论

#51楼[楼主]

@ china-Andy

终于找到了问题的原因，我也不得不说一下，精度是有问题的，应该使用float或者double而不是int。另外你的代码问题太多了，亟待提高。在你原来的程序里面，直接将int换成float即可，下面是我修改的一个稍微精简的版本，仅供参考。

```
1 void RotateVec2( float * px, float * py, float angle
2 {
3     float x = *px - cx;
4     float y = *py - cy;
5
6     *px = x * cosf(angle) - y * sinf(angle) + cx;
7     *py = x * sinf(angle) + y * cosf(angle) + cy;
8 }
9
10 //-----
11 DWORD WINAPI ThreadFunc( LPVOID lpParam )
```

```

12 {
13     RECT rec ;
14     HWND hw = *(HWND*)(lpParam);
15     HDC hdc = GetDC( hw );
16
17     GetClientRect( hw, &rec );
18
19     int i = 0;
20     float fAngle = PI / 60 ;
21     float x = 260 , y = 180; // start point
22     float cx = (rec.right + rec.left) / 2, cy = (rec.b
23
24     DWORD dwColor[2] = { RGB( 255, 0, 0 ),RGB( 0,255,0
25
26     while( 1 )
27     {
28         RotateVec2( &x, &y, fAngle, cx, cy );
29         SetPixel( hdc, x, y, ((i++ / 60) % 2 == 0 ? dwColor[0] : dwColor[1] );
30         Sleep( 50 );
31     }
32     return 0;
33 }

```

支持(0) 反对(0)

2012-09-22 23:08 | zdd

#52楼

LZ，我也申请了博客了，地址是

<http://www.cnblogs.com/star-park/>

对了，你说的精度我也考虑了，不过，SetPixel这个函数不是要求int类型吗？

支持(0) 反对(0)

2012-09-24 14:32 | china-Andy

#53楼[楼主]

@ china-Andy

引用

LZ，我也申请了博客了，地址是

<http://www.cnblogs.com/star-park/>

对了，你说的精度我也考虑了，不过，SetPixel这个函数不是要求int类型吗？

欢迎加入博客园！

SetPixel是要int类型，但是传float进去程序也能运行。使用float或double进行计算，调用SetPixel函数时小数部分会被舍弃，这样虽然会有一定精度损失，但是每个点损失的都一样，所以并不影响最终结果。如果直接用int运算，用你这种方法，每一个点的位置是在前一个点的基础上计算出来的。所以每计算一次就丢失一定的精度，导致越靠后的点损失越严重，最终就是螺旋形状。所以你的方法必须使用float或者double进行计算。

支持(0) 反对(0)

2012-09-24 14:43 | zdd

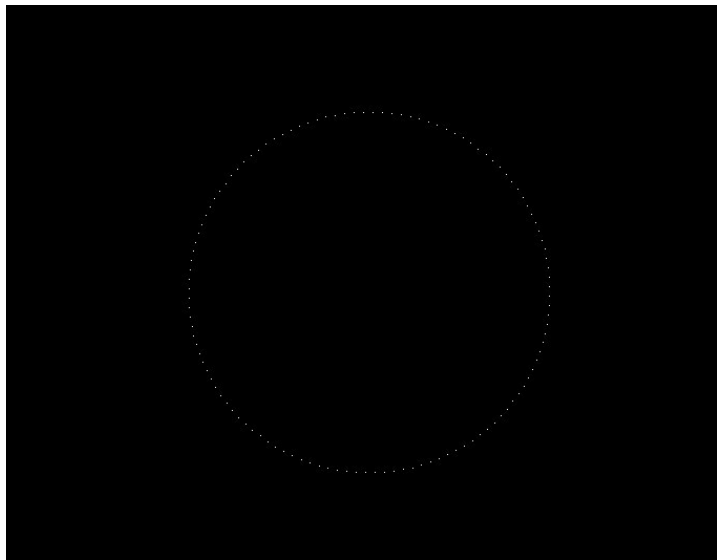
#54楼

虽然，还没试，但是，我想这应该是问题所在，谢谢了，

支持(0) 反对(0)

2012-09-24 14:46 | china-Andy

#55楼



谢谢，谢谢，真的就是精度的问题，这个float参数在画点的时候可以四舍五入，但是计算的时候不能损失精度啊，我真的没想到这方面。

支持(0) 反对(0)

2012-09-24 14:58 | china-Andy

#56楼

楼主，你说，用DX开发2D的游戏，图形程序，一般都有哪些途径！使用什么函数或接口

支持(0) 反对(0)

2012-09-24 15:17 | china-Andy

#57楼[楼主]

@china-Andy

引用

谢谢，谢谢，真的就是精度的问题，这个float参数在画点的时候可以四舍五入，但是计算的时候不能损失精度啊，我真的没想到这方面。

不客气，只要问题解决了就好。不过将float传递给int型参数不是四舍五入，而是截断取整。

支持(0) 反对(0)

2012-09-24 16:58 | zdd

#58楼[楼主]

@ china-Andy

引用

楼主，你说，用DX开发2D的游戏，图形程序，一般都有哪些途径！使用什么函数或接口

用DX开发2D游戏，你可以使用Direct2D来试试。你开发的是什么类型的游戏？

支持(0) 反对(0)

2012-09-24 17:00 | zdd

#59楼

网上有人说，用dx9还不如优化了之后的ddraw，可是根据官方说的意思，以后的dx3d的3d,2d都是用它开发的，不知道当前的游戏公司是怎么选择的。

我想要开发的是单机的格斗游戏，不是类似拳皇的，而是单机版的dnf，但是重点是竞技性，而不是华丽的特效，使用火柴人作为前期的角色模型。

支持(0) 反对(0)

2012-09-24 20:34 | china-Andy

#60楼[楼主]

@ china-Andy

引用

网上有人说，用dx9还不如优化了之后的ddraw，可是根据官方说的意思，以后的dx3d的3d,2d都是用它开发的，不知道当前的游戏公司是怎么选择的。

我想要开发的是单机的格斗游戏，不是类似拳皇的，而是单机版的dnf，但是重点是竞技性，而不是华丽的特效，使用火柴人作为前期的角色模型。

ddraw是老技术了，现在已经完全融合到D3D中，而近两年微软又推出了D2D，DWrite等技术，我觉得还是学习新技术比较好。因为技术的更新换代是不可避免的，就像XP逐渐要被Win7 Win8淘汰一样。

关于你设计的游戏，我觉得很有难度。你有美工一起合作么？如果没有，自己光写代码，没有美工那是很痛苦的。

支持(0) 反对(0)

2012-09-24 20:39 | zdd

#61楼

d2d只支持vista以上系统，我的是xp

支持(0) 反对(0)

2012-09-24 21:24 | china-Andy

#62楼[楼主]

@ china-Andy

引用

d2d只支持vista以上系统，我的是xp

那你就装个Win7啦，Win8都出来了，你还用XP，太落伍了。

支持(0) 反对(0)

2012-09-24 21:27 | zdd

#63楼

我电脑配置不高，怕是撑不起来那个系统啊。

支持(0) 反对(0)

2012-09-24 21:38 | china-Andy

#64楼[楼主]

@ china-Andy

[引用](#)

我电脑配置不高，怕是撑不起来那个系统啊。

这样啊，那你就只能先用XP了，不过没关系，你可以先学学3D，等你换了好电脑再搞D2D，要学的东西很多。总是学不完的。

支持(0) 反对(0)

2012-09-24 21:41 | zdd

#65楼

XP挺不错的啊，我用了10多年了。我才不会跟着微软的鼻子做它的跟屁虫呢。

支持(0) 反对(0)

2013-05-01 22:01 | 蓝杰稀饭

#66楼[楼主]

@ 蓝杰稀饭

[引用](#)

XP挺不错的啊，我用了10多年了。我才不会跟着微软的鼻子做它的跟屁虫呢。

你想用什么是你的自由。

支持(0) 反对(0)

2013-05-01 22:24 | zdd

#67楼

很好的资料，谢谢楼主分享

支持(0) 反对(0)

2013-12-08 11:21 | Stanley0614

#68楼[楼主]

@ Stanley0614

[引用](#)

很好的资料，谢谢楼主分享

多谢支持！

支持(0) 反对(0)

2013-12-09 09:58 | zdd

#69楼

@ zdd

引入齐次矩阵的目的是为了合并矩阵加法和乘法运算，好巧妙

支持(0) 反对(0)

2014-05-09 22:57 | cncyber

#70楼

难得一见的好文！！！！一释我多日忧愁。。。数学老师没把学费退给我啊

支持(0) 反对(0)

2014-06-27 18:02 | yulinxx

#71楼[楼主]

@ cncyber

[引用](#)

@zdd

引入齐次矩阵的目的是为了合并矩阵加法和乘法运算，好巧妙

感谢支持。

支持(0) 反对(0)

2014-06-28 22:41 | zdd

#72楼[楼主]

@ yulinx

引用

难得一见的好文！！！！一释我多日忧愁。。。数学老师没把学费退给我啊

能帮到您，我很欣慰。

支持(0) 反对(0)

2014-06-28 22:41 | zdd

#73楼

我三年的思考，在一瞬间开花了，但是对于绕任意轴旋转，我想破脑筋了还是没明白推到过程。在3d数学基础图形与游戏开发中，放到一个面中推导，甚是费解。

支持(0) 反对(0)

2014-08-28 17:13 | RedLight

#74楼[楼主]

@ RedLight

引用

我三年的思考，在一瞬间开花了，但是对于绕任意轴旋转，我想破脑筋了还是没明白推到过程。在3d数学基础图形与游戏开发中，放到一个面中推导，甚是费解。

关于绕任意轴旋转，请看这里。

<http://www.cnblogs.com/graphics/archive/2012/08/10/2627458.html>

支持(0) 反对(0)

2014-08-29 20:02 | zdd

#75楼

directx不是右手坐标系吗

支持(0) 反对(0)

#76楼[楼主]

@ winning11

DirectX是左手系。

支持(0) 反对(0)

2015-02-11 21:06 | zdd

< Prev 1 2

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云 - 豆果美食、Faceu等亿级APP都在用

【推荐】怎样将“在线Excel”嵌入你的开发系统中？

【推荐】阿里云高性能云服务器+SSD云盘，让业务响应0延迟

野狗API应用于各种实时通信场景

网络快！响应快！开发快！



最新IT新闻：

- 设计师如何跟开发打好关系
 - 罗永浩：手机市场已严重同质化 但锤子科技可借机爆发
 - 精益企业原则之：以产品为中心，而非交付项目
 - 最新研究：电脑病毒或可通过声音进行传播
 - 美国将启动寨卡疫苗效力试验
- » 更多新闻...



最新知识库文章：

- 架构漫谈（九）：理清技术、业务和架构的关系
- 架构漫谈（八）：从架构的角度看如何写好代码
- 架构漫谈（七）：不要空设架构师这个职位，给他实权

- [架构漫谈（六）：软件架构到底是要解决什么问题？](#)
 - [架构漫谈（五）：什么是软件](#)
 - » [更多知识库文章...](#)
-

Powered by:
[博客园](#)
Copyright © zdd