

D3DX库中用D3DXPLANE类来表示平面



|  |
| --- |
| **typedef struct D3DXPLANE**  **{**  **#ifdef \_\_cplusplus**  **public:**  **D3DXPLANE() {}**  **D3DXPLANE( CONST FLOAT\* );**  **D3DXPLANE( CONST D3DXFLOAT16\* );**  **D3DXPLANE( FLOAT a, FLOAT b, FLOAT c, FLOAT d );**  **// casting**  **operator FLOAT\* ();**  **operator CONST FLOAT\* () const;**  **// assignment operators**  **D3DXPLANE& operator \*= ( FLOAT );**  **D3DXPLANE& operator /= ( FLOAT );**  **// unary operators**  **D3DXPLANE operator + () const;**  **D3DXPLANE operator - () const;**  **// binary operators**  **D3DXPLANE operator \* ( FLOAT ) const;**  **D3DXPLANE operator / ( FLOAT ) const;**  **friend D3DXPLANE operator \* ( FLOAT, CONST D3DXPLANE& );**  **BOOL operator == ( CONST D3DXPLANE& ) const;**  **BOOL operator != ( CONST D3DXPLANE& ) const;**  **#endif //\_\_cplusplus**  **FLOAT a, b, c, d;**  **} D3DXPLANE, \*LPD3DXPLANE;**  //a,b,c可以构成平面的法向量n，d是方程：  n.p+d=0,d=-n.p0中的常量d |

## 点和平面的空间关系

给定一个平面（n,d）和点p,

如果n.p+d=0,则点p位于平面上；

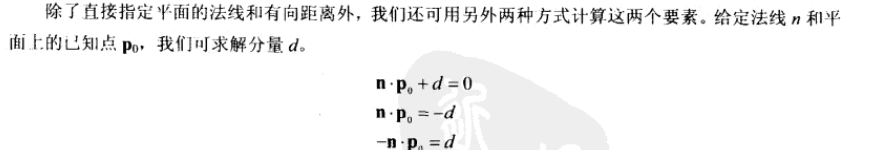
如果n.p+d>0,则点p位于平面前方，并且位于平面的正半区

如果n.p+d<0,则点p位于平面后方，并且位于平面的负半区

D3DX库计算n.p+d的函数

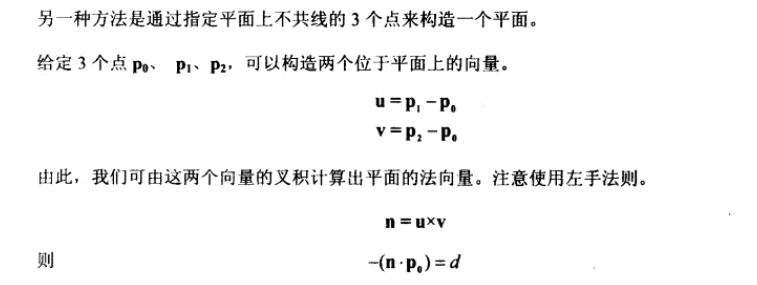
|  |
| --- |
| // ax + by + cz + d，点和平面之见的关系.  **FLOAT D3DXPlaneDotCoord( CONST D3DXPLANE \*pP, CONST D3DXVECTOR3 \*pV);**  **// pP平面的指针，pV点的指针**  **// ax + by + cz + dw** 平面方程与4D向量的点乘.  **FLOAT D3DXPlaneDot( CONST D3DXPLANE \*pP, CONST D3DXVECTOR4 \*pV);**  **// ax + by + cz，**平面与3D向量的点乘  **FLOAT D3DXPlaneDotNormal**  **( CONST D3DXPLANE \*pP, CONST D3DXVECTOR3 \*pV);** |

## 平面的创建



D3DX库创建平面函数

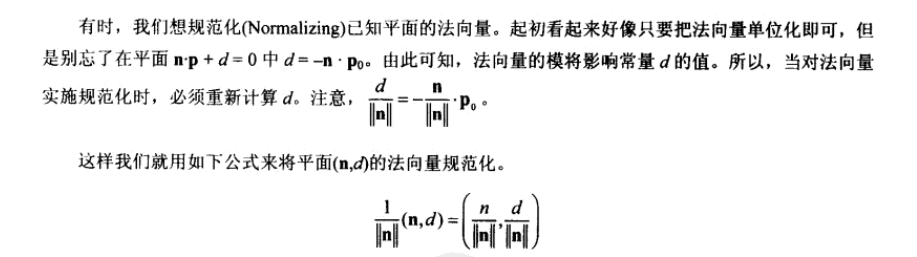
|  |
| --- |
| // Construct a plane from a point and a normal  D3DXPLANE\* WINAPI D3DXPlaneFromPointNormal( D3DXPLANE \*pOut, CONST D3DXVECTOR3 \*pPoint, CONST D3DXVECTOR3 \*pNormal);  //pOut保存结果，pPoint是点的指针，pNormal是平面法向量 |



D3DX库利用三个点创建平面函数

|  |
| --- |
| // Construct a plane from 3 points  **D3DXPLANE\* WINAPI D3DXPlaneFromPoints( D3DXPLANE \*pOut, CONST D3DXVECTOR3 \*pV1, CONST D3DXVECTOR3 \*pV2,CONST D3DXVECTOR3 \*pV3);** |

## 平面的规范化



D3DX库实现平面规范化的函数

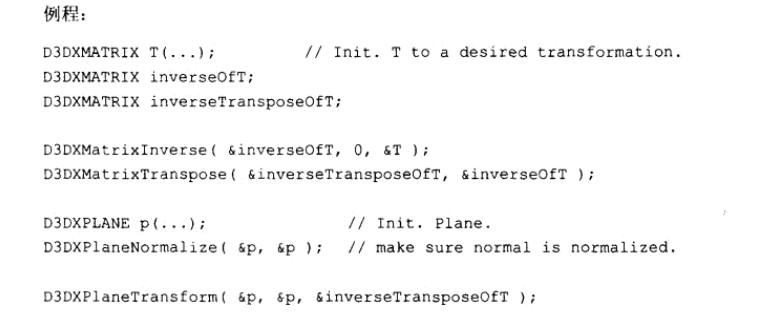
|  |
| --- |
| // Normalize plane (so that |a,b,c| == 1)  D3DXPLANE\* WINAPI D3DXPlaneNormalize( D3DXPLANE \*pOut, CONST D3DXPLANE \*pP);  //pOut接收结果的平面指针，pP需要规范化的平面指针 |

## 平面的变换

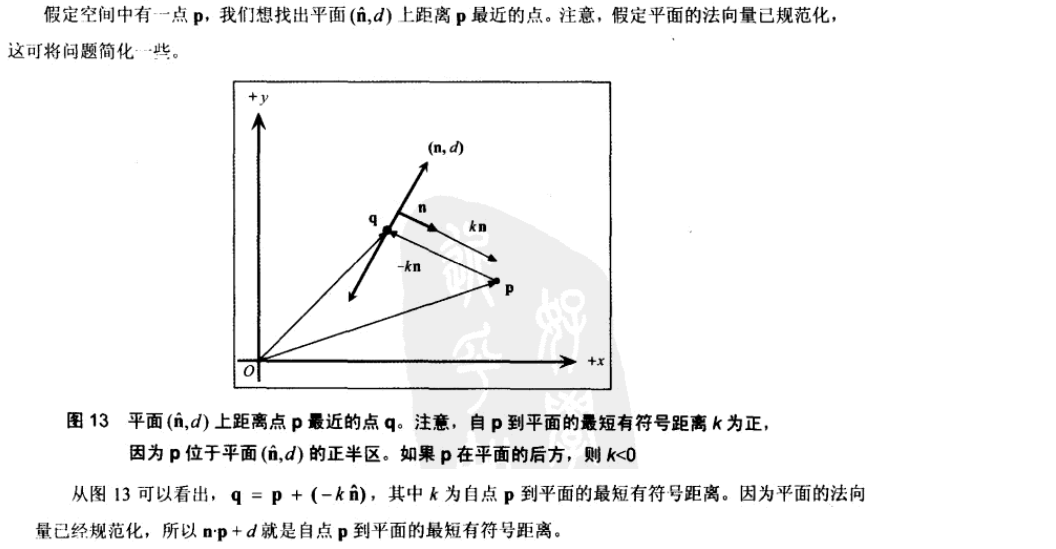


D3DX库实现平面变换的函数

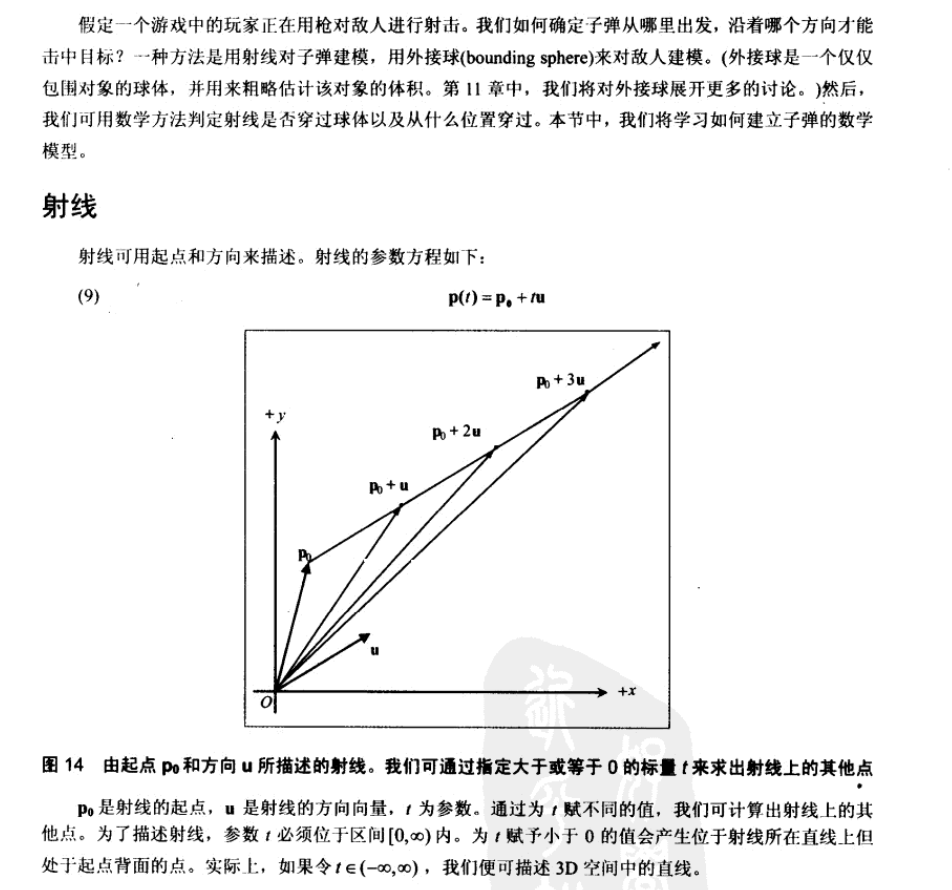
|  |
| --- |
| // Transform a plane by a matrix. The vector (a,b,c) must be normal.  // M should be the inverse transpose of the transformation desired.  D3DXPLANE\* WINAPI D3DXPlaneTransform  ( D3DXPLANE \*pOut, CONST D3DXPLANE \*pP, CONST D3DXMATRIX \*pM );  //pOut接收结果的平面指针，pP需要变换的平面指针，pM变换矩阵 |



## 平面中到某一点的最近点



# 射线



## 射线于平面相交

