

Re: newbie: rotating meshes

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.graphics/2004-07/051>

From: Callum Winter (*callum_at_REMOVE_THISwinter9999.fsnet.co.uk*)

Date: 07/22/04

Date: Thu, 22 Jul 2004 13:29:53 +0100

Hi zodiaq,

Ok I now see.... you want to know how to rotate about an arbitrary axis so if the ship isnt pointing exactly down the z axis and you roll it, it rolls about its own local z instead of the world z, the same going for x and y axis.

OK there are 2 methods, the second i personally prefer using.

Method 1)

First your model should store its own local matrix, and each time you update this matrix you should extract its orientation vectors.

Heres a quick function i wrote to make this easy,

NOTE: m_mLocal should be substituted for your models local matrix.

```
void GetOrientation(D3DXVECTOR3 *vRight, D3DXVECTOR3 *vUp, D3DXVECTOR3
*vDir)
{
    *vRight = D3DXVECTOR3(m_mLocal._11, m_mLocal._21, m_mLocal._31);
    *vUp = D3DXVECTOR3(m_mLocal._12, m_mLocal._22, m_mLocal._32);
    *vDir = D3DXVECTOR3(m_mLocal._13, m_mLocal._23, m_mLocal._33);
}
```

You can then pass any of these vectors to D3DXMatrixRotationAxis, so you can rotate about each one.

Method 2)

Your model should keep a local model matrix and a rotation vector i.e.

D3DXMATRIX m_mLocal, D3DXVECTOR3 m_vRotation.

Make sure to id the local matrix and zero the vector to begin with.

Here are the calls to rotate your ship about its local axis

```
Rotate(0.0f, 0.0f, -fAngle); // Roll Left
```

```
Rotate(0.0f, 0.0f, fAngle); // Roll Right
```

```
Rotate(0.0f, -fAngle, 0.0f) // Yaw Left
```

```
Rotate(0.0f, fAngle, 0.0f) // Yaw Right
```

```
Rotate(-fAngle, 0.0f, 0.0f) // Pitch Up
```

```
Rotate(fAngle, 0.0f, 0.0f) // Pitch Down

// Heres the actual function to build the local ships matrix
void Rotate(float x, float y, float z)
{
    m_vRotation.x += x;
    m_vRotation.y += y;
    m_vRotation.z += z;

    D3DXMATRIX mRotX, mRotY, mRotZ, mRotation;
    D3DXMatrixRotationX(&mRotX, -m_vRotation.x);
    D3DXMatrixRotationY(&mRotY, -m_vRotation.y);
    D3DXMatrixRotationZ(&mRotZ, -m_vRotation.z);
    mRotation = mRotZ * mRotY * mRotX;

    // OK here if you are only using rotation do this
    D3DXMatrixIdentity(m_mLocal);
    D3DXMatrixMultiply(&m_mLocal, &m_mLocal, &mRotation);

    // If you also wanted to incorporate translation do this instead,
    //plugging in your translation matrix.
    // D3DXMatrixMultiply(&m_mLocal, &m_mTrans, &mRotation);
}
```

You will notice in this method the model keeps track of its last rotation vector, so you can simply tell it to rotate n degrees around its own x y or z and it will know how to build the correct matrix, this gives you the effect of rotating the model on its local axis that you want.

Something else you need to know is that rotations in DX are in radians not degrees.

2PI Radians is equal to 360Degrees.,which means 360 degrees in radians is roughly 6.28. because PI is 3.14.

If you divide 6.28 by 360 the resulting value can be multiplied by a number in degrees to convert it to a number in radians.

See if that helps.

Callum.

"Bryce Bangerter" <BryceBangerter@discussions.microsoft.com> wrote in message news:3A3B09B1-3570-47FD-8ACC-6DFE55FFAEC8@microsoft.com...

> *If I understand what you are trying to do correctly, what you want is to use quaternions. Quaternions allow you to do a rotation about an arbitrary axis in 3d space.*

>

> "Zodiaq" wrote:

>

>> *Dnia Wed, 21 Jul 2004 17:12:49 +0100, Callum Winter popelni(a):*

>>

>>> *First position your object in world space at 0,0,0 then rotate it then*

>>> *translate to its appropriate position.*

> > >
> > > *Trying to rotate after a translation will stop the effect where it looks like its rotating around its local axis as in reality every rotates around the origin of the world.*
> > > *In this case positioning at the origin to rotate allows the effect the object is rotating about its own axis, then you just have to shift it to its new position with a translation matrix, after the rotation has been performed.*
> > *Damn! I knew I haven't explain this as I should... first: I'm only using rotation – no translation (moving) or scaling, and object is located in center of world. Second let's say I have some kind of ship, it looks like this:*
> > ^
> > /\
> > <- +
> > V
> > v
> > *where <- is nose (front), ^ and V are wings, and + is engine... Nose is pointed toward negative Z-axis, left wing exactly toward positive x-axis, Y is up...*
> > *Now rotation... using keyboard I rotate this ship, but the it is important order in which I make rotations... in my app first comes rotation around Z, so my ship always rolls as it should... but others rotations doesn't work as I want... I made graphical examples because I find this hard to explain...*
> > *first how rotation are done in DirectX:*
> > http://www.astercity.net/~piotreq/dx_rotations.jpg 8.6kB
> > *now what I currently have:*
> > http://www.astercity.net/~piotreq/dx_what_I_have.jpg 6.9kB
> > *and what I want to have:*
> > http://www.astercity.net/~piotreq/dx_what_I_want.jpg 6.1kB
> > *so my main goal is this:*
> > http://www.astercity.net/~piotreq/dx_what_I_really_want.jpg 6.3kB
> > *help help help! :>*
> > --
> > *...:-Dad, do you think there's people on other planets?*
∴

microsoft.public.win32.programmer.directx.graphics: Re: newbie: rotating meshes

> > ..: *-I don't know, Sparks. But I guess I'd say if it is just us...seems like:.*
> > ..: *an awful waste of space. : O2: zodiac@o2.pl W3:*
<http://www.zodiac.w.pl> :.
> >