

Sprites with Microsoft

Another powerful D3DX component is the ID3DXSprite interface. It provides a set of methods that simplify the process of drawing and manipulating sprites.

If you are going to use sprites, which you probably are then you need to check this interface out. Especially if you are trying to decipher someone's code, it will make things a lot clearer.

The Sprite Model that is used in Games Sprite Models that are used by commercialised game engines are usually based on a set of specifications.

We are using DirectX 9.0 without the summer update. One of the main things to change with the summer update was the ID3DXSprite Interface and its methods such as Draw(). Therefore when referencing the MSDN be careful to find the correct methods and interfaces – usually DirectX 9.0 C++ Archived Content

ID3DXSprite Interface

The ID3DXSprite interface provides a set of methods that simplify the process of drawing sprites using Microsoft Direct3D.

ID3DXSprite Members

`Begin` Prepares a device for drawing sprites.

`Draw` Adds a sprite to the list of batched sprites.

`End` Calls `ID3DXSprite::Flush` and restores the device state to how it was

`Before` `ID3DXSprite::Begin` was called

`Flush` Forces all batched sprites to be submitted to the device. Device states remain as they were after the last call to `ID3DXSprite::Begin`. The list of batched sprites is then cleared

`GetDevice` Retrieves the device associated with the sprite object.

`GetTransform` Gets the sprite transform.

`OnLostDevice` Releases all references to video memory resources and deletes all stateblocks.

`OnResetDevice` Should be called after the device has been reset.

`SetTransform` Sets the sprite transform.

`SetWorldViewLH` Sets the left-handed world-view transform for a sprite. A call to this method is required before billboarding or sorting sprites.

`SetWorldViewRH` Sets the right-handed world-view transform for a sprite. A call to this method is required before billboarding or sorting sprites.

Remarks

The `ID3DXSprite` interface is obtained by calling the `D3DXCreateSprite` function.

The application typically first calls `ID3DXSprite::Begin`, which allows control over the device render state, alpha blending, and sprite transformation and sorting. Then for each sprite to be displayed, call `ID3DXSprite::Draw`. `ID3DXSprite::Draw` can be called repeatedly to store any number of sprites. To display the batched sprites to the device, call `ID3DXSprite::End` or `ID3DXSprite::Flush`.

The `LPD3DXSPRITE` type is defined as a pointer to the `ID3DXSprite` interface.

```
typedef interface ID3DXSprite ID3DXSprite;  
typedef interface ID3DXSprite *LPD3DXSPRITE;
```