Blending(블렌딩)

블렌딩은 전적으로 그리는 순서에 의존한다!

이거때문에 하루종일 삽질

버퍼에 픽셀을 그리면서 후면 버퍼의 값과 섞는 것

두 개의 픽셀 값을 섞는 데는 다음의 식이 이용된다

outputPixel = SourcePixel \* SourceBlendFactor + DestPixel \* DestBlendFactor

각각의 변수는 4D 컬러 벡터 (r,g,b,a)이며, \* 기호는 성분 단위 곱을 나타낸다

outputPixel - 블렌드 결과 픽셀

SourcePixel - 후면 버퍼의 픽셀과 블렌드할 현재 픽셀

SourceBlendFactor - 블렌드할 원본 픽셀의 퍼센트를 지정하는 [0, 1] 범위 내의 값

DestPixel - 후면 버퍼 내의 픽셀

DestBlendFactor - 블렌드할 목적지 픽셀의 퍼센트를 지정하는 [0, 1] 범위 내의 값

블렌딩은 디폴트가 비활성

device->SetRenderState(D3DRS\_ALPHABLENDENABLE, true);

블렌드 인수

D3DRS\_SRCBLEND - SOURCE BLEND 인수

device->SetRenderState(D3DRS\_SRCBLEND, D3DBLEND\_ZERO);

D3DRS\_DESTBLEND - DEST BLEND 인수

device->SetRenderState(D3DRS\_DESTBLEND, D3DBLEND\_ZERO);

블렌드 인수 값은 D3D9 책 182페이지 참조

투명

블렌딩에서 알파값을 이용한 투명을 구현하려면

원본(소스) 블렌드 인수를 D3DBLEND\_SRCALPHA로,

목적지 블렌드 인수를 D3DBLEND\_INVSRCALPHA로 지정해야 한다.

이 값은 디폴트 블렌드 인수이기도 하다.

알파 채널

알파 성분을 보관하기 위해 각 텍셀에 할당된 비트이다

텍스처가 알파 채널을 가지고 있는 경우에는 알파 채널에서 알파를 가져오고,

없다면 버텍스 컬러에서 알파를 얻는다

아래의 렌더 상태를 이용하면 어떤 원본을 이용할 것인지 지정할 수 있다.

// 셰이드 과정에서 난반사 컬러 알파를 계산한다

device->SetTextureStageState(0, D3DTSS\_ALPHAARG11, D3DTA\_DIFFUSE);

device->SetTextureStageState(0, D3DTSS\_ALPHAOP, D3DTOP\_SELECTARG1);

// 알파 채널에서 알파를 얻는다

device->SetTextureStageState(0, D3DTSS\_ALPHAARG1, D3DTA\_TEXTURE);

device->SetTextureStageState(0, D3DTSS\_ALPHAOP, D3DTOP\_SELECTARG1);

Stencil(스텐실)

Stencil Buffer - 특수한 효과를 위한 오프스크린 버퍼로, 후면 버퍼 및 깊이 버퍼와 동일한 해상도를 가짐. 이름이 의미하듯이 스텐실 버퍼는 후면 버퍼의 일정 부분이 렌더링되는 것을 막는 스텐실 효과에 이용된다

스텐실 버퍼를 이용하려면 D3D를 초기화하는 시점에 스텐실 버퍼를 요청해야 하며, 이용할 때 이를 활성화시켜야 한다. 아래와 같이 사용할 수 있다

device->SetRenderState)D3DRS\_STENCILENABLE, true);

// 스텐실 관련 작업

device->SetRenderState)D3DRS\_STENCILENABLE, false);

스텐실 버퍼를 디폴트 값으로 되돌리기 위해서는

device->Clear(0,0,D3DCLEAR\_TARGET | D3DCLEAR\_ZBUFFER | D3DCLEAR\_STENCIL, 0xff000000, 1.0f, 0);

세번째 인자로 D3DCLEAR\_STENCIL을 넘겨주면 된다

스텐실 버퍼는 깊이 버퍼를 만들 때 함께 만들 수 있으며, 깊이 버퍼 포맷을 지정할 때 아래와 같이 함께 지정할 수 있다.

D3DFMT\_D24S8 - 32비트 깊이/스텐실 버퍼를 만들어 깊이에 24비트 스텐실에 8비트를 할당

D3DFMT\_D24X4S4 - 32비트 깊이/스텐실 버퍼를 만들어 깊이에 24비트 스텐실에 4비트, 나머지 4비트는 사용하지 않음

D3DFMT\_D15S1 - 16비트 깊이/스텐실 버퍼를 만들어 깊이에 15비트 스텐실에 1비트를 할당

스텐실 테스트는 아래와 같다

(ref & mask) cmp (val & mask)

스텐실이 활성화되어 있다는 가정 하에 모든 픽셀에 대해 스텐실 테스트가 수행되고, 이 때 두 개의 피연산자를 이용한다.

LHS = ref & mask - 애플리케이션이 정의한 스텐실 참조 값과 애플리케이션이 정의한 마스크 값의 AND 연산으로 얻어진다

RHS = val & mask - 현재 테스트하려는 픽셀의 스텐실 버퍼(값)와 애플리케이션이 정의한 마스크 값의 AND 연산으로 얻어진다

여기에 비교 연산자에 지정된 방법으로 LHS와 RHS를 비교하는 스텐실 테스트가 수행되며, 표현식은 부울 값으로 평가된다. 식이 true(통과)면 후면 버퍼의 픽셀을 출력하며, false(실패)면 픽셀이 출력되는 것을 막는다. 후면 버퍼에 픽셀이 쓰여지지 않으면 깊이 버퍼에도 또한 쓰여지지 않는다

스텐실 테스트에 이용되는 변수는 사용자가 제어가 가능하다

D3DRS\_STENCILREF(스텐실 참조값) – 기본 0

Device->SetRenderState(D3DRS\_STENCILREF,0x1);

D3DRS\_STENCILMASK(스텐실 마스크값) – 기본 0xffffffff

Device->SetRenderState(D3DRS\_STENCILMASK,0x0000ffff);

스텐실 값 – 스텐실 테스팅을 수행하고 있는 현재 픽셀의 스텐실 버퍼 값이다. ij번째 픽셀을 테스트하고 있다면 이 값은 스텐실 버퍼의 ij번째 값이 된다.

D3DRS\_STENCILFUNC(비교 연산자) 아래의 목록 중 하나를 사용하면 된다.

typedef enum \_D3DCMPFUNC {

D3DCMP\_NEVER = 1, - 항상 실패

D3DCMP\_LESS = 2, - LHS < RHS

D3DCMP\_EQUAL = 3, - LHS = RHS

D3DCMP\_LESSEQUAL = 4, - LHS <= RHS

D3DCMP\_GREATER = 5, - LHS > RHS

D3DCMP\_NOTEQUAL = 6, - LHS != RHS

D3DCMP\_GREATEREQUAL = 7, - LHS >= RHS

D3DCMP\_ALWAYS = 8, - 항상 성공

D3DCMP\_FORCE\_DWORD = 0x7fffffff,

} D3DCMPFUNC;

스텐실 버퍼 갱신하기

특정한 픽셀이 후면 버퍼에 쓰여질지의 여부를 결정하는 것 이외에도 다음과 같은 세 가지의 가능한 상황에 따라 스텐실 버퍼 항목이 갱신되는 방법을 정의할 수 있다.

Device->SetRenderState(D3DRS\_STENCILFAIL, StencilOperation); - 스텐실 테스트 실패

Device->SetRenderState(D3DRS\_STENCILZFAIL, StencilOperation); - 깊이 테스트 실패

Device->SetRenderState(D3DRS\_STENCILPASS, StencilOperation); - 깊이/스텐실 테스트 성공

StencilOperation 값은 아래 중 하나를 넣는다

D3DSTENCILOP\_KEEP – 스텐실 버퍼 항목을 변경하지 않음.(현재 유지)

D3DSTENCILOP\_ZERO – 스텐실 버퍼 항목을 0으로 지정

D3DSTENCILOP\_REPLACE – 스텐실 버퍼 항목을 스텐실 참조 값으로 대체

D3DSTENCILOP\_INCRSAT – 스텐실 버퍼 항목을 증가. 최대 값을 넘으면 최대 값으로 고정

D3DSTENCILOP\_DECRSAT – 스텐실 버퍼 항목을 감소. 최소 값을 넘으면 0으로 고정

D3DSTENCILOP\_INVERT – 스텐실 버퍼 항목을 반전.

D3DSTENCILOP\_INCR – 스텐실 버퍼 항목을 증가, 최대를 넘으면 0으로 고정

D3DSTENCILOP\_DECR – 스텐실 버퍼 항목을 감소, 0을 넘으면 최대로 고정

스텐실 쓰기 마스크

스텐실 버퍼에 쓰여지는 모든 값을 마스크하는 쓰기 마스크를 설정할 수 있다. 디폴트는 0xffffffff 아래는 상위 16비트를 마스크하는 예이다

Device->SetRenderState(D3DRS\_STENCILWRITEMASK, 0x0000ffff);