

Font

305890
2009년 봄학기
5/6/2009
박경신

Overview

- ID3DXFont 인터페이스를 이용해 텍스트를 렌더링 하는 방법
- CD3DFont 클래스를 이용해 텍스트를 렌더링 하는 방법
- 초당 렌더링 되는 프레임 수 (fps)를 계산하는 방법
- D3DXCreateText 함수를 이용해 3D 텍스트를 만들고 렌더링 하는 방법

ID3DXFont

- 글꼴 출력 방법
 - ID3DXFont - 내부적으로 GDI를 이용. 복잡한 글꼴과 포맷을 지원함.
 - CD3DFont - GDI가 아닌 Direct3D를 이용. 매우 빠름. 간단한 텍스트 출력에 적합함.

ID3DXFont

- ID3DXFont 인터페이스 - 텍스트를 그리는데 사용

```
HRESULT D3DXCreateFontIndirect (  
    LPDIRECT3DDEVICE9 pDevice,  
    CONST D3DXFONT_DESC *pDesc,  
    LPD3DFONT *ppFont);
```

- pDevice - 글꼴과 연결된 장치
- pDesc - 글꼴을 나타내는 D3DXFONT_DESC 구조체. 만들고자 하는 글꼴의 특성을 지정.
- ppFont - 만들어진 글꼴을 return

ID3DXFont

▣ D3DXFONT_DESC 구조체

```
typedef struct D3DXFONT_DESC {
    UINT Height;        UINT Width; // 폰트문자의 높이와 너비
    UNIT Weight;        // 굵기 단위, 범위 0 (없음) ~ 1000 (두꺼움)
    UINT MipLevels;     // mip level의 수
    BOOL Italic;        // true이면 이탤릭체로 나타남
    BYTE CharSet;       // 캐릭터 셋 지정
    BYTE OutputPrecision; // output이 요청한 사양과 얼마나
                        // 부합되어야 하는지 명시
    BYTE Quality;       // output quality
    BYTE PitchAndFamily; // pitch and family of the font
    TCHAR FaceName[LF_FACESIZE]; // 32문자 이내로 이름 지정
} D3DXFONT_DESC;
```

ID3DXFont

▣ D3DXCreateFontIndirect 이용 방법 예제

```
ID3DXFont *Font = 0;
D3DXFONT_DESC fontDesc;
ZeroMemory(&fontDesc, sizeof(D3DXFONT_DESC));
fontDesc.Height = 25; // in logical units
fontDesc.Width = 12; // in logical units
fontDesc.Weight = 500; // boldness, range 0(light) - 1000(bold)
fontDesc.MipLevels = D3DX_DEFAULT;
fontDesc.Italic = false;
fontDesc.CharSet = DEFAULT_CHARSET;
fontDesc.OutputPrecision = 0;
fontDesc.Quality = 0;
fontDesc.PitchAndFamily = 0;
strcpy(fontDesc.FaceName, "Times New Roman"); // font style
if (FAILED(D3DXCreateFontIndirect(Device, &fontDesc, &Font))) {
    ::MessageBox(0, "D3DXCreateFontIndirect() - FAILED", 0, 0);
    ::PostQuitMessage(0);
}
```

Text Drawing

▣ ID3DXFont 인터페이스 - 텍스트를 그리는데 사용

```
INT ID3DXFont::DrawText (
    LPD3DXSPRITE pSprite,
    LPCTSTR pString, INT Count, LPRECT pRect,
    DWORD Format, D3DCOLOR Color);
```

- ▣ pSprite - string을 가지는 ID3DXSprite 객체. NULL일 경우 Direct3D가 자신의 sprite object으로 렌더링 함.
- ▣ pString - 그리고자 하는 string을 가리키는 포인터
- ▣ Count - String의 문자의 수. -1인 경우 string이 null로 끝난다고 가정하고 문자수를 자동 계산한다.
- ▣ pRect - Text가 그려질 화면 상의 영역을 지정하는 RECT 구조체를 가리키는 포인터
- ▣ Format - Text format을 지정하는 선택적 플래그. DT_CENTER 등.
- ▣ Color - 텍스트 색

Text Drawing

▣ DrawText의 사용 예제:

```
Font->DrawText(NULL,
    "Hello World",
    -1, // size of string or -1 indicates null terminating string
    &rect, // rectangle text is to be formatted to in windows coords
    DT_TOP | DT_LEFT, // draw in the top left corner of the viewport
    0xff000000); // black text

// draw FPS text
std::stringstream FPSStr;
FPSStr << "FPS: " << CalcFPS(timeDelta);
RECT rect = {5, 5, width, height};
Font->DrawText(0, FPSStr.str().c_str(), -1, &rect, DT_TOP, DT_LEFT,
    0xffffffff);
```

Calculating FPS

- FPS (frames rendered per second) - 초당 렌더링 되는 프레임 수로 매 초마다 FPS를 계산한다.

```
DWORD FrameCnt = 0;
float TimeElapsed = 0;
float FPS = 0;
void CalcFPS(float timeDelta) {
    FrameCnt++;
    TimeElapsed += timeDelta;

    if(TimeElapsed >= 1.0f) {
        FPS = (float)FrameCnt / TimeElapsed;

        TimeElapsed = 0.0f;
        FrameCnt = 0;
    }
}
```

Example: ID3DXFont

```
#include "u3dUtility.h"
ID3DXFont* Font = 0;
DWORD FrameCnt = 0;
float TimeElapsed = 0;
float FPS = 0;
char FPSString[9];
bool Setup() {
    D3DXFONT_DESC fontDesc;
    ZeroMemory(&fontDesc, sizeof(D3DXFONT_DESC));
    fontDesc.Height = 25;    fontDesc.Width = 12;
    fontDesc.Weight = 500;   fontDesc.MipLevels = D3DX_DEFAULT;
    fontDesc.Italic = false; fontDesc.CharSet = DEFAULT_CHARSET;
    fontDesc.OutputPrecision = 0;    fontDesc.Quality = 0;
    fontDesc.PitchAndFamily = 0;
    strcpy(fontDesc.FaceName, "Times New Roman"); // font style
    if (FAILED(D3DXCreateFontIndirect(Device, &fontDesc, &Font))) {
        ::MessageBox(0, "D3DXCreateFontIndirect() = FAILED", 0, 0);
        ::PostQuitMessage(0);
    }
    return true;
}
```

Example: ID3DXFont

```
bool Display(float timeDelta) {
    if (Device) {
        FrameCnt++;
        TimeElapsed += timeDelta;
        if (TimeElapsed >= 1.0f) {
            FPS = (float)FrameCnt/TimeElapsed;
            sprintf(FPSString, "%f", FPS);
            FPSString[8] = '\0'; // mark the end of string
            TimeElapsed = 0.0f;
            FrameCnt = 0;
        }
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0xffffffff, 1.0, 0);
        Device->BeginScene();
        RECT rect = {0, 0, Width, Height};
        Font->DrawText(NULL, FPSString, -1, &rect, DT_TOP | DT_LEFT, 0xff000000);
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    }
    return true;
}
```

Example: ID3DXFont



CD3DFont

■ CD3DFont class

- DirectX SDK는 매우 유용한 유틸리티 코드 제공
- DXSDK\Samples\C++\Common
- Direct3D와 텍스처를 입힌 삼각형을 이용해 텍스트를 렌더링
- ID3DFont에 비해 훨씬 빠른 속도
- 복잡한 글꼴과 포매팅은 지원 못함
- 간단한 텍스트를 빠른 속도로 그리고자 할 때 사용

■ 다음 파일들이 응용 프로그램에 추가되어야 함:

- d3dfont.h, d3dfont.cpp, d3dutil.h, d3dutil.cpp, dxutil.h, dxutil.cpp

CD3DFont

■ CD3DFont 구성하기

CD3DFont(const TCHAR *strFontName, DWORD dwHeight,
DWORD dwFlags=0L);

- strFontName - 글꼴 이름 (null로 끝나는 문자열)
- dwHeight - 글꼴의 높이
- dwFlags - 선택적 생성 플래그 : 0 또는 D3DFONT_BOLD, D3DFONT_ITALIC, D3DFONT_ZENABLE 플래그 조합

■ CD3DFont 객체 생성 후 글꼴을 초기화

```
CD3DFont* Font = 0;  
Font = new CD3DFont("Times New Roman", 16, 0);  
Font->InitDeviceObjects( Device );  
Font->RestoreDeviceObjects();
```

CD3DFont

■ Text Drawing

HRESULT CD3DFont::DrawText(
Float x,
Float y,
DWORD dwColor,
const TCHAR *strText,
DWORD dwFlags=0L);

- x, y - 텍스트 그리기를 시작한 x, y 좌표
- dwColor - 텍스트 컬러
- strText - 그리고자 하는 문자열의 포인터
- dwFlags - 선택적 렌더링 플래그 : D3DFONT_CENTERED, D3DFONT_TWOSIDED, D3DFONT_FILTERED 플래그 조합

```
Font->DrawText(20, 20, 0xff000000, "Hello, World");
```

CD3DFont

■ CD3DFont 객체를 삭제하기 전에, 다음과 같은 정리 루틴을 호출해서 마무리한다.

```
Font->InvalidateDeviceObjects();  
Font->DeleteDeviceObjects();  
Delete Font;
```

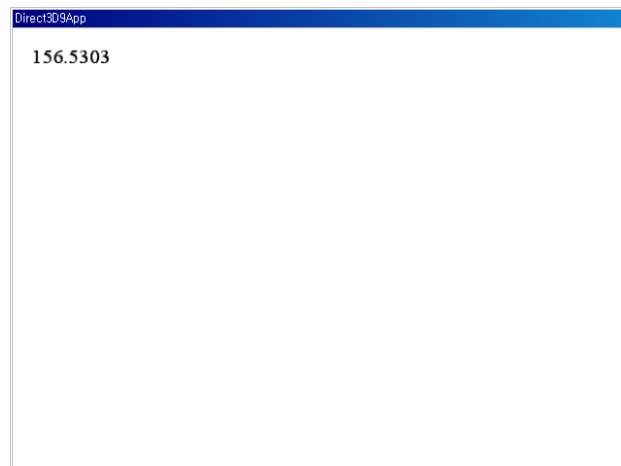
Example: CD3DFont

```
#include "d3dUtility.h"
#include "d3dfont.h"
CD3DFont* Font = 0;
DWORD FrameCnt = 0;
float TimeElapsed = 0;
float FPS = 0;
char FPSString[9];
bool Setup() { // create a font object and initialize it
    Font = new CD3DFont("Times New Roman", 16, 0);
    Font->InitDeviveObject(Device);
    Font->RestoreDeviceObjects();
    return true;
}
void cleanup() {
    if (Font) {
        Font->InvalidateDeviceObjects();
        Font->DeleteDeviceObjects();
        d3d::Delete<CD3DFont*> (Font);
    }
}
```

Example: CD3DFont

```
bool Display(float timeDelta) {
    if (Device) {
        FrameCnt++;
        TimeElapsed += timeDelta;
        if (TimeElapsed >= 1.0f) {
            FPS = (float)FrameCnt/TimeElapsed;
            sprintf(FPSString, "%f", FPS);
            FPSString[8] = '\0'; // mark the end of string
            TimeElapsed = 0.0f;
            FrameCnt = 0;
        }
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0xffffffff, 1.0, 0);
        Device->BeginScene();
        if (Font)
            Font->DrawText(20, 20, 0xff000000, FPSString);
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    } return true;
}
```

Example: CD3DFont



D3DXCreateText

■ 텍스트의 3D Mesh 생성하고 이를 렌더링 함.

```
HRESULT D3DXCreateText(
    LPDIRECT3DDEVICE9 pDevice, HDC hDC,
    LPCTSTR pText,
    FLOAT Deviation, FLOAT Extrusion,
    LPD3DXMESH* ppMesh, LPD3DXBUFFER* ppAdjacency,
    LPGLYPHMETRICSFLOAT pGlyphMetrics);
```

- pDevice - 메쉬와 연결될 장치
- hDC - 메쉬 생성하는 데 이용될 글꼴 정보를 포함하는 장치 컨텍스트의 핸들
- pText - 메쉬로 만들 텍스트를 포함하는 null로 끝나는 text string
- Deviation - 0 이상의 값. 0 이면 원본 글꼴의 한 디자인 단위와 같음
- Extrusion - z축 방향으로 켄 글꼴의 깊이
- ppMesh - 만들어진 mesh
- ppAdjacency - 만들어진 mesh의 adjacency 정보. 필요하지 않으면 NULL
- pGlyphMetrics - string의 문자수 만큼의 glyph. 필요하지 않으면 NULL

Example: D3DXCreateText

```
ID3DXMesh* Text = 0;
bool Setup() { // create a font object and initialize it
    HDC hdc = CreateCompatibleDC(0);
    HFONT hFont; HFONT hFontOld;
    LOGFONT lf;
    ZeroMemory(&lf, sizeof(LOGFONT));
    lf.lfHeight = 25; lf.lfWidth = 12;
    lf.lfEscapement = 0; lf.lfItalic = false;
    lf.lfUnderline = false; lf.lfStrikeOut = false;
    lf.lfCharSet = DEFAULT_CHARSET;
    lf.lfOutPrecision = 0; lf.lfClipPrecision = 0;
    lf.lfQuality = 0; lf.lfPitchAndFamily = 0;
    strcpy(lf.lfFaceName, "Times New Roman");
    // create font and select it with the device context
    hFont = CreateFontIndirect(&lf);
    hFontOld = (HFONT) SelectObject(hdc, hFont);
    // create text mesh
    D3DXCreateText(Device, hdc, "Direct3D", 0.001f, 0.4f, &Text, 0, 0);
}
```

Example: D3DXCreateText

```
// restore the old font and free the acquired HDC
SelectObject(hdc, hFontOld);
DelectObject(hFont);
DeleteDC(hdc);

// light
// camera
// projection matrix
return true;
}
```

Example: D3DXCreateText

```
bool Display(float timeDelta) {
    if (Device) {
        D3DXMATRIX yRot, T;
        static float y = 0.0f;
        D3DXMatrixRotationY(&yRot, y);
        y += timeDelta;
        if (y >= 6.28f) y = 0.0f;
        D3DXMatrixTranslation(&T, -1.6f, 0.0f, 0.0f);
        T = T * yRot;
        Device->SetTransform(D3DTS_WORLD, &T);
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0x00000000, 1.0, 0);
        Device->BeginScene();
        Device->SetMaterial(&d3d::WHITE_MTRL);
        Text->DrawSubset(0);
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    } return true;
}
```

Example: D3DXCreateText

