

DirectXTK 나눔고딕 SpriteFont 생성 가이드

DirectXTK에서 나눔고딕 한글 폰트를 사용하기 위한 .spritefont 파일 생성 방법을 단계별로 설명합니다.

☰ 목차

1. 사전 준비
2. MakeSpriteFont.exe 다운로드
3. .spritefont 파일 생성
4. 프로젝트에서 사용하기
5. 트러블슈팅

사전 준비

1. 나눔고딕 폰트 설치 확인

Windows에 나눔고딕 폰트가 설치되어 있는지 확인합니다.

```
# Windows Fonts 폴더에서 확인  
dir C:\Windows\Fonts\*Nanum*.ttf
```

예상 출력:

```
NanumGothic.ttf  
NanumGothicBold.ttf
```

만약 설치되어 있지 않다면:

- [네이버 나눔글꼴 공식 사이트](#)에서 다운로드
- Windows용: [NanumFontSetup_TTF_GOTHICCALL.exe](#) 실행하여 설치

MakeSpriteFont.exe 다운로드

DirectXTK의 MakeSpriteFont 도구를 다운로드합니다.

방법 1: GitHub Releases에서 직접 다운로드 (권장)

```
# 프로젝트 폴더로 이동  
cd e:\강의\clifedge\CPPGP\Projects\CPPGP2025  
  
# MakeSpriteFont.exe 다운로드  
curl -L -o MakeSpriteFont.exe  
https://github.com/microsoft/DirectXTK/releases/download/oct2025/MakeSpriteFont.exe
```

방법 2: DirectXTK 소스 빌드

DirectXTK 저장소를 클론하고 `DirectXTK/Desktop_2019.sln` 또는 최신 솔루션 파일을 빌드하면 `MakeSpriteFont\bin\Release\MakeSpriteFont.exe` 가 생성됩니다.

.spritefont 파일 생성

1. 기본 ASCII 문자만 포함 (영문, 숫자, 기호)

```
./MakeSpriteFont.exe NanumGothic NanumGothic_16.spritefont
```

결과:

- 파일명: `NanumGothic_16.spritefont`
- 파일 크기: 약 40KB
- 포함 문자: 95개 글리프 (기본 ASCII)
- 사용 사례: 영문만 표시하는 경우

출력 예시:

```
Importing NanumGothic
Captured 95 glyphs
Cropping glyph borders
Packing glyphs into sprite sheet
Packing efficiency 94.39664%
Premultiplying alpha
Writing NanumGothic_16.spritefont (CompressedMono format)
```

2. 한글 완성형 전체 포함 (가-횡) ★ 추천

```
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Korean_16.spritefont
/CharacterRegion:0xAC00-0xD7A3 /FastPack"
```

파라미터 설명:

- `NanumGothic` : Windows에 설치된 폰트명
- `NanumGothic_Korean_16.spritefont` : 출력 파일명
- `/CharacterRegion:0xAC00-0xD7A3` : 한글 완성형 유니코드 범위 (가-횡)
- `/FastPack` : 빠른 패킹 알고리즘 사용 (대용량 문자셋에 필수)

결과:

- 파일명: `NanumGothic_Korean_16.spritefont`
- 파일 크기: 약 11MB
- 포함 문자: 11,172개 글리프 (한글 완성형 전체)
- 패킹 효율: 약 90%
- 요구사항: Direct3D Feature Level 9.3 이상

출력 예시:

```
Importing NanumGothic
.....
Captured 11172 glyphs
Cropping glyph borders
Packing glyphs into sprite sheet
Packing efficiency 90.12913%
WARNING: Resulting texture requires a Feature Level 9.3 or later device.
Premultiplying alpha
Writing NanumGothic_Korean_16.spritefont (CompressedMono format)
```

3. 영문 + 한글 모두 포함

```
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Full_16.spritefont /CharacterRegion:0x0020-
0x007E /CharacterRegion:0xAC00-0xD7A3 /FastPack"
```

파라미터 설명:

- /CharacterRegion:0x0020–0x007E : 기본 ASCII (영문, 숫자, 기호)
- /CharacterRegion:0xAC00–0xD7A3 : 한글 완성형 (가-힣)

주요 명령줄 옵션

문자 범위 지정

```
/CharacterRegion:0xAC00–0xD7A3 # 한글 완성형 (가-힣)
/CharacterRegion:0x0020–0x007E # 기본 ASCII
/CharacterRegion:0x3131–0x3163 # 한글 자음/모음
```

폰트 스타일

```
/FontSize:16 # 폰트 크기 (기본값: 23)
/FontStyle:Bold # Regular, Bold, Italic, Strikeout, Underline
/Sharp # 선명한 안티앨리어싱
```

최적화

```
/FastPack # 빠른 패킹 (대용량 문자셋 필수)
/TextureFormat:CompressedMono # 텍스처 포맷 지정
```

프로젝트에서 사용하기

ZFont 클래스에서 사용 예제

```
// 헤더 파일 (ZFont.h)
#include "SpriteFont.h"
#include "SpriteBatch.h"

class ZFont {
private:
    DirectX::SpriteFont* m_pSpriteFont;
    DirectX::SpriteBatch* m_pSpriteBatch;

public:
    BOOL Create(ZGraphics& gfx, const char* SpriteFontFile, long Size = 16,
                BOOL Bold = FALSE, BOOL Italic = FALSE);
    BOOL FastPrint(long XPos, long YPos, const char* text);
};

// 구현 파일 (ZFont.cpp)
BOOL ZFont::Create(ZGraphics& gfx, const char* SpriteFontFile, long Size,
                    BOOL Bold, BOOL Italic)
{
    if (SpriteFontFile == NULL)
        return FALSE;

    ID3D11Device* pDevice = gfx.GetDeviceCOM();
    ID3D11DeviceContext* pContext = gfx.GetDeviceContext();

    if (pDevice == NULL || pContext == NULL)
        return FALSE;

    try
    {
        // char* to wstring 변환
        std::wstring wSpriteFontFile(SpriteFontFile,
                                      SpriteFontFile + strlen(SpriteFontFile));

        // DirectXTK SpriteFont와 SpriteBatch 생성
        m_pSpriteFont = new DirectX::SpriteFont(pDevice, wSpriteFontFile.c_str());
    }
}
```

```

    m_pSpriteBatch = new DirectX::SpriteBatch(pContext);

    return TRUE;
}
catch (...)
{
    return FALSE;
}
}

BOOL ZFont::FastPrint(long XPos, long YPos, const char* text)
{
    if (m_pSpriteFont == nullptr || m_pSpriteBatch == nullptr)
        return FALSE;

    // char* to wstring 변환
    std::wstring wText(text, text + strlen(text));

    // DirectXTK SpriteFont 사용
    m_pSpriteBatch->Begin();
    DirectX::XMFLOAT2 pos((float)XPos, (float)YPos);
    m_pSpriteFont->DrawString(m_pSpriteBatch, wText.c_str(), pos,
                               DirectX::Colors::White);
    m_pSpriteBatch->End();

    return TRUE;
}

```

사용 예제

```

// 폰트 객체 생성
ZFont font;

// 영문 폰트 로드
font.Create(graphics, "NanumGothic_16.spritefont", 16);

// 또는 한글 폰트 로드
font.Create(graphics, "NanumGothic_Korean_16.spritefont", 16);

// 텍스트 출력
font.FastPrint(100, 100, "안녕하세요!");
font.FastPrint(100, 120, "Hello World!");

```

트러블슈팅

문제 1: "Too many arguments" 오류

원인: 명령줄 파싱 문제 (Bash vs PowerShell)

해결방법:

```

# ❌ 잘못된 방법 (Bash에서)
./MakeSpriteFont.exe NanumGothic output.spritefont /CharacterRegion:0xAC00-0xD7A3

# ✅ 올바른 방법 (PowerShell 사용)
powershell -Command ".\MakeSpriteFont.exe NanumGothic output.spritefont /CharacterRegion:0xAC00-0xD7A3"

```

문제 2: 텍스처 크기 경고

경고 메시지:

```
WARNING: Resulting texture requires a Feature Level 9.3 or later device.
```

설명:

- 한글 전체를 포함하면 텍스처가 매우 커집니다
- Direct3D Feature Level 9.3 이상의 GPU 필요 (대부분의 현대 GPU는 지원)

해결방법:

1. `/FastPack` 옵션 사용 (이미 사용 중)
2. 필요한 문자만 선택적으로 포함
3. 폰트 크기 줄이기 (`/FontSize:14`)

문제 3: 파일 크기가 너무 큼

현상: 한글 폰트 파일이 11MB로 큼

최적화 방법:

방법 1: 실제 사용하는 문자만 포함

게임/앱에서 사용하는 모든 텍스트를 분석하여 실제 사용하는 한글 글자만 추출

```
# 예: "가나다라마바사아자차카타파하"만 포함
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Custom.spritefont /CharacterRegion:0xAC00-0xB098 /CharacterRegion:0xB2E4 /FastPack"
```

방법 2: 자주 사용하는 한글만 포함

```
# 가-나, 다-라, 마-바 등 일부 범위만 포함
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Partial.spritefont /CharacterRegion:0xAC00-0xB098 /FastPack"
```

방법 3: 폰트 크기 줄이기

```
# 14pt 폰트 사용
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Korean_14.spritefont /FontSize:14 /CharacterRegion:0xAC00-0xD7A3 /FastPack"
```

문제 4: 한글이 깨져서 표시됨

원인: char 대신 `wchar_t` 또는 `wstring` 사용 필요

해결방법:

```
// ❌ 잘못된 방법
font.FastPrint(100, 100, "안녕하세요");

// ✅ 올바른 방법 - UTF-8 문자열을 wstring으로 변환
std::string utf8Text = "안녕하세요";
std::wstring wText(utf8Text.begin(), utf8Text.end());
// 또는 코드에서 자동 변환 구현
```

유니코드 범위 참고표

범위	설명	유니코드
기본 ASCII	영문, 숫자, 기호	0x0020-0x007E
한글 완성형	가-횡	0xAC00-0xD7A3
한글 자음	ㄱ-ㅎ	0x3131-0x314E
한글 모음	ㅏ-ㅣ	0x314F-0x3163

범위	설명	유니코드
숫자 (전각)	0 - 9	0xFF10-0xFF19
특수문자	! " # \$ %...	0xFF01-0xFF5E

참고 자료

- DirectXTK Wiki - MakeSpriteFont
- DirectXTK Wiki - SpriteFont
- DirectXTK GitHub Repository
- 네이버 나눔글꼴

생성 결과 요약

성공적으로 생성된 파일

파일명	크기	글리프 수	용도
NanumGothic_16.spritefont	40KB	95	영문 전용
NanumGothic_Korean_16.spritefont	11MB	11,172	한글 완성형 전체

생성 시간

- 영문 폰트: 약 1초
- 한글 폰트: 약 2-3분 (FastPack 사용)

최종 명령어

```
# 1. MakeSpriteFont.exe 다운로드
curl -L -o MakeSpriteFont.exe
https://github.com/microsoft/DirectXTK/releases/download/oct2025/MakeSpriteFont.exe

# 2. 영문 폰트 생성
./MakeSpriteFont.exe NanumGothic NanumGothic_16.spritefont

# 3. 한글 폰트 생성 (FastPack 사용)
powershell -Command ".\MakeSpriteFont.exe NanumGothic NanumGothic_Korean_16.spritefont
/CharacterRegion:0xAC00-0xD7A3 /FastPack"
```

작성일: 2025-10-31

DirectXTK 버전: 2025.7.10.1

폰트: 나눔고딕 (NanumGothic.ttf)