

Mobile Device를 위한 3D GUI Tool Frame 개발 및 적용

The Application and Development of 3D GUI Tool Frame for the Mobile Devices

송영한, 이경숙

(주)새론즈, 대구산업정보대학 컴퓨터정보계열
인하대학교 컴퓨터정보학부
yhsong@swmobile.co.kr, kslee@tpic.ac.kr

Song Young-Han, Lee Kyung-Sook

Saerons Co., LTD,
Daegu Polytechnic College University

요약

본 논문에서는 다양한 휴대형 단말기의 플랫폼 친화적이며 콘텐츠 개발의 신속성, 확장성 및 용이성이 고려된 3D 콘텐츠 플랫폼 엔진과 Application Frame 기술을 Convergence하였으며, Mobile Device에 탑재 가능한 3D 그래픽 사용자 인터페이스(GUI)의 시제품을 제작 및 응용하고, Mobile Game 등 콘텐츠의 개발을 효과적으로 빠르고 쉽게 구현할 수 있는데 필요한 3D GUI 객체(widget) 및 툴킷을 포함하는 Frame을 표준화 하는데 기본이 되는 연구를 하였다.

I. 서론

다양하고 복잡한 기능을 가진 모바일 기기의 출현에 따라 2D의 평면적이고 단순화 된 형태의 GUI로는 그 기능을 정확히 표현할 수 없는 한계 및 제약이 따르므로 최근에는 3D 개념의 입체적인 GUI를 도입하여 사용자의 인지능력 향상과 편의성을 극대화하는 추세이다. 따라서 모바일 기기에서도 3D GUI 객체(widget)를 효과적으로 빠르고 쉽게 구현할 수 있는 SDK환경과 3G GUI 개발 환경 제공에 대한 연구가 필요하게 되었다.

근래에 이르러 3D 그래픽 처리를 위한 모바일 기기용 가속 하드웨어 장치가 일반화 되는 추세이고 몇몇 국내 업체를 비롯하여 다양한 가속칩이 모바일 기기에 장착되게 되었다.

이에 따라 모바일 기기용 3차원 그래픽 사용자 인터페이스(3D GUI)의 표준 개발 방법 연구 및 보편화 된 3D 게임엔진 기술을 활용하여 모바일 기기에 탑재 가능한 3차원 그래픽 사용자 인터페이스(GUI)를 제작하고, 이의 제작을 효과적으로 빠르고 쉽게 구현할 수 있는데

필요한 3D GUI 객체 (widget) 및 툴킷을 포함하는 SDK의 표준화가 필요하며, 또한 상용화 된 다양한 모바일 기기(Mobile Dvice)에 대한 연구가 필요하다.

본 논문에서는 PC에서 3D GUI 디자인 작업이 가능한 모바일용 3D GUI Engine기술을 기본으로 하여 다양한 기기의 플랫폼에 적용할 수 있는 Frame 개념의 Tool을 융합(Convergence)하여 보았다.

따라서 본문의 첫번째는 PC에서 모바일 기기용 3D GUI 설계 작업이 가능한 EUI-Builder의 개념 이해와 EUI-Builder에서 작업된 3D GUI의 모바일 기기에 적용 가능한 소스로 자동 변환 기능을 적용하였고, 두번째는 모바일 기기에 적합하도록 경량화 된 모바일 기기용 3D GUI Engine인 EUI-Engine을 응용 개발하였다. 세번째는 EUI-Builder와 EUI-Engine을 활용하여 모바일 기기의 메뉴 시스템을 시제작 하였으며, 효율성 및 차후 추가 연구 되어야 할 여러 문제점에 대하여 논의하고자 한다.

우리나라의 경우 유례없는 다양한 형태의 모바일 단말기 보급률과 세계적으로 인정받는 고성능 모바일 단

말기 생산 능력을 보유하고 있는 실정이며, 모바일 시장의 빠른 성장 속도와 국내 업체의 역량을 감안해 볼 때, 모바일 기기와 사용자간의 인터페이스인 3D GUI 분야의 전망은 매우 밝다고 할 수 있다.

II. 3D GUI Tool Frame 개념 및 적용기술

본 논문에서 이용된 기술은 2006년 한국전자통신연구원으로부터 이전 받은 차세대 온라인 게임 S/W 기술 개발 과제에서 모바일/PC/콘솔의 연동을 위한 게임엔진 기술 및 시범 콘텐츠 제작을 위해 개발되었으며, 타 플랫폼의 게임엔진과의 연동을 위한 기술을 포함한 고성능 모바일 기기에 최적화 된 엔진이다.

2-1. Direct 3D 모바일의 개요

Direct 3D 모바일(Microsoft Direct 3D Mobile)은 윈도우 CE 5.0, 윈도우 임베디드 CE 6.0에 제공되는 API이다. PC용 DirectX API에 비하여 기능이 축소되어 있지만, 현재는 많은 임베디드 기기에서 사용되는 표준 3D API로 자리 잡고 있다.

2-2. Direct 3D 모바일의 주요 기능 및 개발 환경

현재 출시되는 ARM 111 코어 기반 마이크로프로세서에는 2D, 3D 그래픽 가속칩이 탑재되어 있다. Centrality의 Tatian 칩, NXP, 삼성 및 국내 중소기업인 넥서스칩에서도 그래픽 가속 칩을 출시하고 있다. Direct 3D 는 소프트웨어로써 3D를 구현하는 API이며, 앞서 기술한 그래픽 가속칩을 사용하기 위해서는 벤더에서 제공하는 전용 드라이브를 사용하게 된다. 본 연구에서도 소프트웨어로 3D 처리를 하도록 하였고 주요 기능은 아래와 같다.

- Flat shading과 Fouraud shading 지원
- 다중 라이트 소스 지원
- 소프트웨어 에물레이션 드라이브 지원
- Transformation과 clipping 지원

- Backbuffer를 사용한 page flipping 지원
- Window mode와 Full screen mode 지원

III. Mobile 3D GUI Tool Frame 기술 개발

Mobile 3D GUI Tool 기술은 제한적인 모바일 환경에 최적화 된 경량화 3D GUI SDK와 저용량 Texture 및 저용량 Effect를 사용하여 3D GUI를 구성하는 다양한 형태의 Widget을 포함하며, 보다 확장성이 용이한 UI 구조와 3D GUI 개발용 3D Resource의 제작 및 통합 관리틀의 기능을 포함하고 있다.

3-1. Mobile 3D GUI Tool Frame의 개요

가. 구성

- ① 모바일 3D GUI 빌더(EUI-Builder): 3D 모델 제작 틀에서 작성된 오브젝트를 로드하여 원하는 형태로 배치, 애니메이션 설정, 카메라 세팅 등의 기능을 수행
- ② 모바일 3D GUI 엔진(EUI-Engine): 모바일 빌더에서 설정된 데이터를 로드하여 실제 모바일 환경 하에서 동작할 수 있게 하는 프로그램
- ③ 샘플 프로그램: ①과 ②로 제작한 메뉴 프로그램

나. 동작환경

- ① 모바일 메뉴 빌더: Microsoft Windows XP Platform
- ② 모바일 엔진: Microsoft Mobile Direct 3D가 지원되는 Windows CE Platform

3-2. Mobile 3D GUI Tool Frame 개발환경

주요 개발환경은 아래와 같으며 주요한 Frame 산출물은 <표 1>에서 부연 설명하였다.

- Microsoft VISUAL STUDIO 2005
- Microsoft Direct X SDK (JUNE 2007)
- Boost STL 1,33,1
- Microsoft Windows Mobile 5.0 Pocket PC SDK
- Microsoft Windows Mobile 5.0 Smartphone SDK

표 1. 주요 산출물구조

| 저장소 | 설명 |
|-------------------------|---|
| 3D_COMMON | DirectX 3D에 관련된 API를 Wrapping한 모바일/PC Cross Platform GUI Engine |
| COMMON | Platform 독립적으로 개발에 관련된 공통 기능을 모아 놓은 소스들 |
| DEBUGGER | 디버그 라이브러리를 사용할 수 있게 하는 인터페이스 소스들 |
| EUIBUILDER | 모바일 메뉴 빌더 프로그램의 주요 소스들 |
| EUIBUILDER/BUILDER_DATA | 모바일 메뉴 빌더 프로그램으로 모바일 엔진을 제작할 때 쓰는 주요 데이터들 |
| BUILD_TEST_1/PC | 모바일 메뉴 빌더로 만든 첫 번째 샘플 프로그램(PC 환경 예제) |
| BUILD_TEST_1/MOBILE | 모바일 메뉴 빌더로 만든 첫 번째 샘플 프로그램(모바일 환경 예제) |
| EUICOMMON | 모바일 엔진과 관련된 공통 소스들 |
| EUIENGINE_MOBILE | 모바일 엔진 소스들(모바일 환경) |
| EUIENGINE_PC | 모바일 엔진 소스들(PC 환경) |
| LIB | 프로젝트에 필요한 라이브러리들 |
| BNTUI | 모바일 메뉴 빌더의 MFC 라이브러리 헤더 |
| PATH | 카메라 세팅과 연관된 인터페이스 소스들 |
| PATHLIB | PC 환경 카메라 세팅 소스들 |
| PATHLIBMOBILE | 모바일 환경 카메라 세팅 소스들 |

3-3. 세부 개발 내용 및 방법

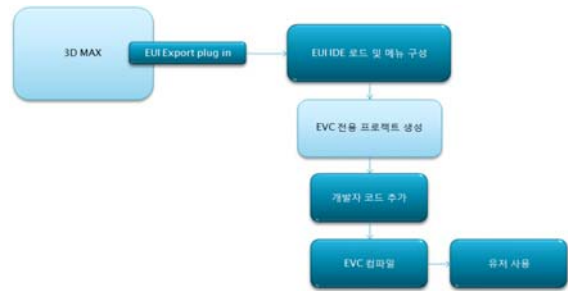
가. 모바일 3D GUI 빌더(EUI-Builder) 프로그램

나. 모바일 프로그램의 개발 순서

- ① 3D 오브젝트 틀에서 오브젝트 생성
- ② EUI-Builder로 오브젝트로드 및 배치 후 프로젝트 생성
- ③ 생성된 프로젝트를 Microsoft Visual studio에서 로드하여 필요 소스 추가 및 컴파일.
- ④ 생성된 모바일 프로그램을 모바일 디바이스에 업로드 및 사용

다. 모바일 개발 프레임 워크

모바일 빌더 프로그램으로 생성된 산출물은 Microsoft 기본 Win32 API 구조를 그대로 가지고 있으며 모바일 엔진은 동적 연결 라이브러리에 보관되어 컴파일된다.

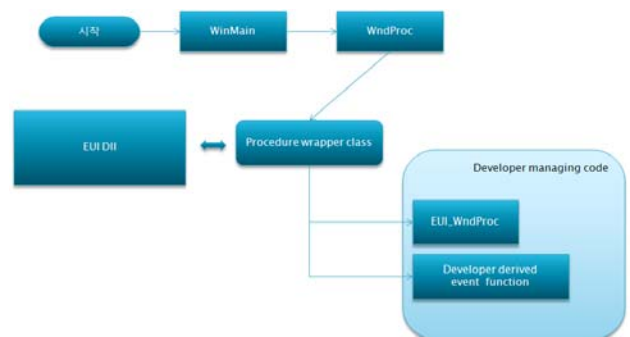


▶▶ 그림 1. 모바일 콘텐츠 개발 순서

[그림 1]와 같이 대부분의 소스는 프로젝트 생성 시 자동으로 기록되고 모바일 개발자는 엔진에서 호출하여 주는 이벤트 함수를 자신에게 맞는 모양으로 수정하여 사용하며, 쉽게 3D 오브젝트를 사용하는 모바일 디바이스 프로그램을 만들 수 있다.

- WinMain: 모든 함수 호출 및 메시지 분배를 담당
- MyRegisterClass: 프로그램 윈도우 클래스를 등록
- InitInstance: 모든 윈도우 관련 기능을 초기화
- WndProc: 메시지 발생 시 시스템에 의해 호출되며, 모든 메시지 처리를 담당한다.
- is_screen_rotate: 모바일 빌더 프로그램에 의해 생성

[그림 2]은 모바일 프로그램과 모바일 엔진의 실행 순서를 보여주고 있다. [그림 3]에서 보는 바와 같이 모바일 프로그램은 모바일 엔진의 초기화를 담당하여야 하며, 모바일 엔진은 관련 이벤트가 발생하면 모바일 프로그램의 함수를 호출한다.



▶▶ 그림 2. 모바일 개발 프레임 워크

라. 모바일 빌더로 산출되는 주요 소스 및 라이브러리

| | |
|--|--|
| EUIInterface.cpp EUIInterface.h | 모바일 엔진을 사용하기 위한 인터페이스 함수 정의 |
| 프로젝트명.cpp | 모바일 빌더로 생성된 프로젝트의 메인소스 |
| EUIEngine_mobile.dll EUIEngine_pc.dll | 모바일 엔진 라이브러리 |
| EUIConfig.EUI | 모바일 빌더로 만들어진 오브젝트 배치/카메라 세팅/기타효과를 정의한 파일 |

마. 모바일 엔진 인터페이스

모바일 엔진의 모든 인터페이스는 EUIInterface.cpp 와 EUIInterface.h에 자동으로 생성되며, 모바일 개발자는 다음의 인터페이스 함수 중 용도에 맞게 on_selected 함수를 수정하여 사용하여야 한다.

3-4. 구현 알고리즘

3D 개발툴에서 생성된 3D 오브젝트를 파싱하는 알고리즘은 3D오브젝트는 Microsoft에서 지원하는 X file 중 텍스트 타입을 기준으로 하며, Microsoft사는 모바일용 X file 라이브러리를 지원하지 않으므로 직접 개발하여 이용하였으며, 메쉬 데이터 로드 및 렌더링, 텍스처링, 애니메이션 기능을 지원하도록 자체적으로 구현하였다.

모바일 빌더는 제작된 3D 오브젝트를 로드하여 모바일 엔진에서 읽을 수 있도록 배치 및 기타 효과를 주는 프로그램으로써 크게 다음과 같이 나눌 수 있다.

가. 오브젝트 파싱 파트 : 3D 오브젝트를 로드하여 화면에 디스플레이하는 기능

나. 메뉴트리 구성 파트 : 로드된 오브젝트를 적합하게 구성

다. 프로젝트 빌드 파트 : 배치된 오브젝트를 모바일 엔진에서 읽을 수 있도록 구성하는 부분

3-5. 3D 오브젝트 클래스

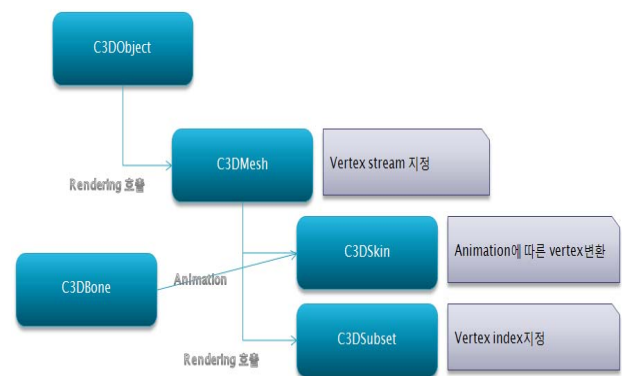
로드된 3D 오브젝트 데이터는 상황에 따라 원본 데이터 클래스와 복사본 데이터 클래스에 맞추어져 분산 저장되며 원본 데이터 클래스는 다음과 같은 특징을 가진다.

가. Mesh 파일에서 실제 데이터를 읽어서 저장하는 객체

나. Mesh, Animation, Bone구조의 원본 데이터를 실제로 가지고 있다.

다. 한번 읽어진 데이터를 프로그램 종료시까지 변하지 않는다.

3-6. 3D 오브젝트 렌더링 및 애니메이션 과정



▶▶ 그림 3. 렌더링 과정

모든 3D 오브젝트는 C3D Object 클래스가 최상위 클래스이며 C3D Object는 관련된 애니메이션 및 렌더링 클래스를 가지고 있다. [그림 4]처럼 애니메이션 과정을 거친 오브젝트는 [그림 3]처럼 모든 Vertex가 애니메이션에 따라 변화하고 변화된 Vertex를 렌더링 한다.

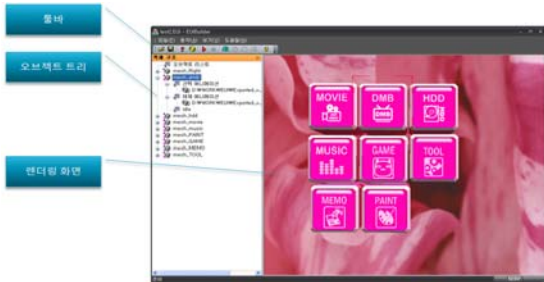


▶▶ 그림 4. 애니메이션 과정

애니메이션은 C3DX Animation_set 객체에 보관되는 C3DX Animation 객체 중 하나를 선택하여 수행하며, 애니메이션 정보는 C3D Bone 객체에 전달되어 C3D Skin 객체에 적용된다.

3-7. 모바일 3D GUI 빌더 프로그램의 주요 클래스

가. 전체 화면



▶▶ 그림 5. 모바일 빌더 전체 화면

모바일 빌더 화면은 [그림 5]와 같이 “실행부”, “오브젝트 트리” 및 “렌더링 화면”의 3부분으로 나뉘며, “실행부”는 모든 실행에 관련된 명령을 할 수 있고, “오브젝트 트리”는 현재 화면에 로드된 오브젝트를 보여준다. “렌더링 화면”은 “오브젝트 트리”에서 로드된 모든 3D 오브젝트를 렌더링하여 보여준다.

나. 3D 오브젝트 로드 및 배치

모바일 빌더는 각 3D 오브젝트 별로 “IDLE”, “선택”, “해제”의 3가지 애니메이션을 지원한다. 각 애니메이션 정보가 꼭 필요하지는 않다. 애니메이션이 지정되지 않으며 애니메이션 없는 모습 그대로를 렌더링한다.

- ① Main 오브젝트 X file
- ② 각 3D 오브젝트에 대한 애니메이션 X file



▶▶ 그림 6. 애니메이션 추출

애니메이션을 로드한 후 개발자는 원하는 애니메이션을 임의로 실행 시킬 수 있다.

다. 모바일 빌더 사용(프로젝트 빌드)

3D 오브젝트를 배치 및 카메라 세팅이 끝났다면 “빌드”를 통하여 실제 모바일 디바이스에 적용할 프로젝트를 생성할 수 있다.

프로젝트가 생성되면 모바일 디바이스 전용 3D 엔진과 PC 전용 3D 엔진 및 엔진을 사용하는 소스가 같이 생성된다.



▶▶ 그림 7. 저장 장소 결정

그리고, 프로젝트를 빌드한 후 어려없이 사용하기 위해서는 모바일 빌더에서 사용한 3D 오브젝트 파일 및 텍스처 파일을 반드시 같은 폴더로 복사시켜야 한다.

IV. 모바일 엔진(Mobile Engine)

4-1. EUI 엔진

모바일 엔진에서 사용되는 알고리즘과 주요 인터페이스에 대해 살펴보면 다음과 같다.

모바일 엔진은 3D 오브젝트와 관련된 모든 기능은 모바일 개발자에게 노출이 되지 않도록 한다. 모바일 프로그램 개발자는 Win CE 프로젝트 개발에 관련된 지식만 가지고 있다면 개발이 가능하도록 하였다.

4-2. 알고리즘 및 주요 소스

가. 초기화

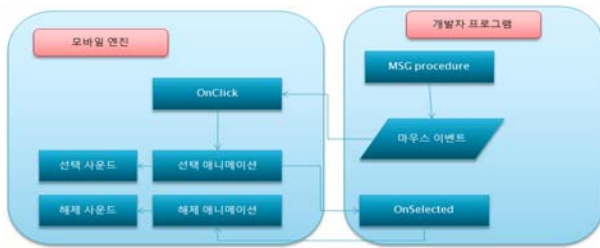
초기화는 WinMain에서 호출하는 InitInstance에서 호출하도록 프로젝트 생성 시 자동으로 삽입된다.

- void reg_sound_on_select(LPCTSTR path)
- void reg_sound_on_deselect(LPCTSTR path)
- path : 사운드 파일의 위치 및 파일 이름

모바일 개발자는 WinMain의 메시지 파싱 루프에서 다음과 같이 idle 인터페이스를 호출하여야 한다. on_idle 인터페이스 함수는 프로세서가 남는 시간을 활용하여 관련된 3D 오브젝트를 렌더링 한다.

나. 선택/해제 이벤트

모바일 프로그램은 메시지 프로시저에서 마우스 혹은 모바일 디바이스 포인터 입력 메시지를 받아서 모바일 엔진 인터페이스의 on_click 함수를 호출하여야 한다. 호출 인자는 3D 오브젝트의 이름과 모바일 엔진 내부에서 사용하는 3D 오브젝트의 id를 넘겨준다. 모바일 프로그램은 id를 사용하여 반드시 deselect 함수를 호출하여야 한다.



▶▶ 그림 8. 선택/해제 시 흐름도

4-3. 사용 방법

개발자는 다음의 간단한 순서에 따라 모바일 디바이스 프로그램을 제작한다.

- 가. 모바일 프로젝트 빌더를 사용하여 프로젝트를 생성
- 나. 모바일 프로젝트 빌더에서 사용하였던 3D오브젝트 파일과 텍스처 파일을 프로젝트 폴더로 복사
- 다. 프로젝트명.vcproj 를 MS사의 Visual studio 2005 를 사용
- 라. 프로젝트명.cpp 의 InitInstance 함수에서 프로그램 이 실행되는 폴더를 정확하게 입력

마. 프로젝트 성격에 따라 InitInstance 함수에 사운드 파일을 등록

바. EUIInterface.cpp에 있는 on_selected 함수를 자신의 용도에 맞게 수정

사. 프로젝트를 컴파일

아. 모바일 디바이스에 업로드 하여 사용

4-5. 적용사례 구현

다음은 Tool Frame을 사용하여 사이드 메뉴 방식과 정면 메뉴 방식으로 디자인하고 구현하여 보았다.

가. 사이드 메뉴 방식



▶▶ 그림 9. 사이드메뉴 GUI 실행 화면

[그림9]에서 보듯이 3D 효과를 잘 보여주며 카메라 이동으로 구현되었다. 3D 오브젝트는 평상시는 위아래로 흔들리는 애니메이션을 가지고 있고, 선택되어지면 앞으로 돌출되며 해제 시 다시 뒤로 들어가 위 아래로 흔들린다. 카메라는 선택되어진 메뉴로 이동된다.

나. 정면 메뉴 방식

배열 형태의 메뉴이며 3D 오브젝트는 선택되어지면 앞으로 돌출되며 해제시 다시 들어간다. 카메라는 선택되어진 메뉴로 이동한다.

V. 결론

현재 개발되고 있는 주종을 이루는 모바일 기기들의 User Interface의 경우 대부분이 2D로 개발되어 있으며 일본의 초기 디자인 경쟁제품을 답습하고 있다. 이러한 문제점은 모바일 개발사들의 제품 UI에 대한 인식 부족과 3D 적용 등으로 인한 개발 단가 상승에 기인한

다. 따라서 본 논문의 결과를 응용하면 국산 3차원 모바일 사용자 인터페이스를 적용하여 국내 개발되는 모바일 기기의 상품 경쟁력을 높이는데 일조할 것으로 기대된다.

또한 WinCE를 기반으로 한 모바일 기기 개발업체와 모바일 기기용 S/W 제작업체를 대상으로 Direct 3D 기반 소프트웨어 기반으로 향후 EUI-Engine 부분에 3D 하드웨어 가속과 관련된 포팅레이어를 추가하여 다양한 개발자의 요구에 신속하게 대응할 수 있도록 EUI-Engine부분의 연구도 계속 되어져야 할 것이다.

본 논문 결과의 주요 활용분야는 휴대폰을 비롯한 모바일 기기용 어플리케이션 개발을 위한 다양한 형태의 모바일 기기에 새로운 형태의 3D GUI를 적용하고 3D GUI Builder를 사용한 자동화 및 최적화 기능을 활용하여 제품경쟁력 상승을 위한 사용자 인터페이스 설계 및 구현에 도움을 줄 것으로 예상된다.

■ 참고 문헌 ■

- [1] ETRI DCRD 2006년 기술이전 자료(모바일 3D게임 엔진)
- [2] 한국산업개발원, “게임엔진 품질평가기술”, 도서출판 정일 2003.
- [3] <http://www.clgames.co.kr>
- [4] <http://www.swmobile.co.kr>
- [5] RanderWare, <http://www.renderware.com>
- [6] OpenGL, <http://www.opengl.org>
- [7] Microsoft DirectX,
<http://www.microsoft.com/direct>
- [8] 김성용, 한정현 3D차원 렌더링:기술적 이슈를 중심으로, 정보처리학회지 2005, 제 12권 제 6호 P24~33
- [9] 김혜선, 크로스플랫폼 게임엔진기술 ETRI Dicon Insight Apr 2006 P52~56
- [10] 정보통신부 한국정보통신교육원 한국컴퓨터 게임학회 “Game 엔진 개발론”, 홍릉과학 출판사 pp.39~53, 2003
- [11] T Welsh, Parallax Mapping with offset limiting: A per-pixel approximation of uneven surfaces, Infiscape Corporation, 2004