

스마트폰 게임 콘텐츠를 위한 GNEX-to-iPhone 변환기의 설계 및 구현

이양선[†], 최효준^{**}, 김재성^{***}

요 약

국내 이동통신사들이 서로 다른 모바일 플랫폼을 채택하여 사용함으로써 개발자는 하나의 모바일 게임 콘텐츠를 서비스하기 위하여 각각의 스마트폰 플랫폼 특성에 맞추어 변환 작업을 하여야 한다. 하지만, 기존의 모바일 게임 콘텐츠를 스마트폰 플랫폼으로 이식하기 위한 변환 작업에 많은 시간과 비용이 소모되고 있다. 이는 다양한 콘텐츠가 제공되지 못하고 있는 원인이기도 하다. 본 논문에서는 이런 문제를 해결하기 위해 GNEX 플랫폼의 모바일 게임 콘텐츠를 iPhone 플랫폼(iOS)의 스마트폰 콘텐츠로 변환해주는 모바일 콘텐츠 변환기 시스템을 구현하였다. GNEX-to-iPhone 모바일 콘텐츠 변환기 시스템은 단시간 내에 아이폰 스마트폰 플랫폼으로 콘텐츠를 이식할 수 있도록 하여 동일 콘텐츠를 다른 이동통신사에 서비스하는데 소모되는 시간과 비용을 최소화해준다. 또한, 기존 콘텐츠를 자동 변환하여 타 플랫폼에 서비스함으로써 콘텐츠의 재사용성을 높이고, 신규 콘텐츠의 생산성을 높여 사용자에게는 다양한 모바일 게임 콘텐츠를 제공할 수 있도록 지원한다.

Design and Implementation of the GNEX-to-iPhone Converter for Smart Phone Game Contents

YangSun Lee[†], HyoJun Choe^{**}, JaeSung Kim^{***}

ABSTRACT

Since Korean mobile communication companies each use different mobile platforms, developers must configure and translate their game contents to run under each of the smart phone platforms so that they can be serviced correctly. Nevertheless, such translation tasks require lengthy times and costs. This is one of the reasons why a variety of contents could not be provided. In order to mitigate such difficulty, this paper implemented the mobile contents converter system that converts mobile game contents of the GNEX platform to smart phone contents of the iPhone platform(iOS). The GNEX-to-iPhone automatic mobile contents translation system helps minimize the amount of time and cost required in servicing contents to different mobile communication companies by promptly translating platform-specific contents to run under iPhone smart phone platforms. Also, the automatic translation and servicing of existing contents increases the reusability of these contents and also the productivity of new contents thereby offering users with a more variety of games.

Key words: Automatic Mobile Contents Converter(모바일 콘텐츠 자동 변환기), Mobile Platform(모바일 플랫폼), Smart Phone Platform(스마트폰 플랫폼), GNEX(지넥스), iPhone(아이폰), Contents Analyzer(콘텐츠 분석기), Platform Mapping Engine(플랫폼 매핑 엔진)

※ 교신저자(Corresponding Author): 이양선, 주소: 서울시 성북구 정릉동 16-1(136-704), 전화: 02)940-7743, FAX: 02)940-7616, E-mail: yslee@skuniv.ac.kr
접수일: 2011년 1월 5일, 수정일: 2011년 2월 16일
완료일: 2011년 3월 19일

[†] 종신회원, 서경대학교 컴퓨터공학과 교수

^{**} 정회원, 서경대학교 컴퓨터공학과 대학원생
(E-mail: tenesia@hanmail.net)

^{***} 준회원, 서경대학교 컴퓨터공학과 대학원생
(E-mail: stoms@lycos.co.kr)

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (No.20100023644).

1. 서론

현재 국내 이동통신사별로 서로 다른 모바일 플랫폼을 채택하여 사용함으로 인해, 모바일 게임 콘텐츠 개발자는 하나의 콘텐츠를 서비스하기 위하여 각 스마트폰 플랫폼의 특성을 고려하여 다양한 버전의 콘텐츠를 중복 개발해야 한다. 이는 개발자들로 하여금 이미 개발된 콘텐츠를 스마트폰 플랫폼으로 이식하는 작업에 대한 필연성을 유발하게 하였다. 하지만, 하나의 모바일 게임 콘텐츠의 소스 및 리소스를 분석하여 다른 스마트폰 플랫폼으로 이식하는 작업에 많은 시간과 비용이 소모된다. 새로운 콘텐츠를 개발하는 것보다는 하나의 콘텐츠를 서비스하기 위해 시간과 비용이 중복 투자되고 있다[1-5].

이런 문제를 해결하기 위해 한 플랫폼의 콘텐츠를 다른 플랫폼에서도 실행할 수 있도록 자동으로 변환해주는 콘텐츠 자동 변환기 시스템을 연구하였다. 콘텐츠 자동 변환기 시스템은 모바일 게임 콘텐츠를 단시간 내에 다른 플랫폼으로 이식할 수 있도록 하여, 동일 콘텐츠를 다른 이동통신사에 서비스하는데 소모되는 인력, 시간, 비용을 절약해준다[6-12].

본 논문에서는 모바일 GNEX 게임 콘텐츠를 아이폰의 스마트폰 게임 콘텐츠로 자동 변환해주는 GNEX-to-iPhone 변환기를 설계하고 구현하였다. 기존의 GNEX 모바일 게임 콘텐츠를 iPhone 콘텐츠로 자동 변환함으로써 기존 콘텐츠의 재사용성을 높이고 스마트폰 사용자가 보다 다양한 콘텐츠를 제공할 수 있도록 하였다.

2. 관련 연구

2.1 GNEX 플랫폼

GNEX 가상기계(virtual machine)는 GNEX 응용 프로그램을 해석하고 실행하는 역할을 한다. GNEX 커널은 다양한 시스템 인터페이스를 제공하며, 메모리 관리자 탑재로 GNEX 시스템을 보고하고 응용 프로그램의 크기 및 힙(heap) 메모리 제약 등을 해소하는 역할을 한다. 이벤트 핸들러는 플랫폼의 이벤트를 받아 GNEX 이벤트로 변환하고 각 이벤트에 대응되는 알고리즘을 호출하여 처리한다. MIDD(Mobile Interface Device Driver)는 사운드 재생, LCD 출력 등 단말기의 하드웨어 관련 기능을 단말기 플랫폼에

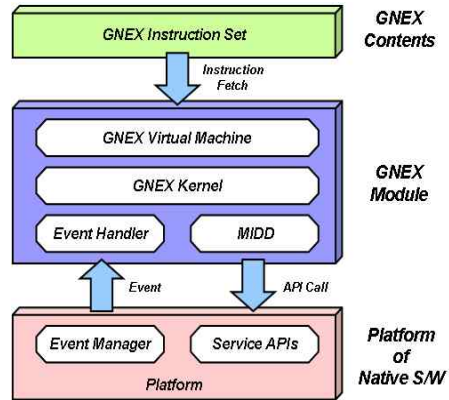


그림 1. GNEX의 구조

서 제공하는 API를 이용하여 구현한 것이며, 플랫폼 API의 호출과 실행 결과를 처리한다.

GNEX 응용 프로그램은 ANSI C 언어를 기반으로 한 모바일 C 언어로 개발한다. GNEX는 VDI (Variable Depth Image)라는 단말기 전용으로 설계된 이미지 형식을 사용한다. VDI는 픽셀 당 할당되는 비트 수를 가변적으로 정의하여 사용하는 일종의 비트맵 형식의 이미지 규격이다. GNEX 응용 프로그램에서는 BMP와 같은 이미지 파일을 GNEX SDK에서 제공하는 이미지 도구를 통해 VDI 형식으로 변환하여 모바일 C 소스 코드에 포함시킨다. 사운드 리소스 역시 GNEX 응용 프로그램에서 사용할 수 있도록 GNEX 규격의 사운드 파일로 변환하여 모바일 C 소스 코드에 포함하여 사용한다[13-15].

2.2 iPhone 플랫폼

iPhone 플랫폼인 iOS(iPhone Operating System)는 애플의 스마트폰인 아이폰과 디지털 미디어 재생 기기인 아이팟 터치, 태블릿형 컴퓨터인 아이패드 내에 내장되어 있는 운영체제이다. 실제로는 Mac OS X를 기반으로 만들어져 있으며 Mac OS X의 기본 구조와 비슷하다. iOS는 하드웨어와 화면에 보이는 어플리케이션의 중간 역할을 한다. 어플리케이션은 직접 하드웨어와 연결되지 않으며 시스템 인터페이스를 통해서 연결된다.

iOS는 코어 OS 계층(core OS layer), 코어 서비스 계층(core services layer), 미디어 계층(media layer), 코코아 터치 계층(cocoa touch layer)의 네 개의 소프트웨어 계층으로 구성되어 있다. 그림 2는 iOS의 구

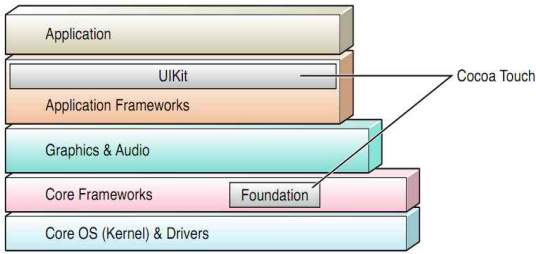


그림 2. iOS 플랫폼의 구조

조를 나타낸 것이다.

코어 OS와 코어 서비스 계층은 기본적인 시스템 API등을 제공한다. iOS는 Mac OS X와 같은 유닉스 기반으로서 POSIX 표준을 준수하는 API들을 제공한다. 네트워크 소켓에 관련된 부분이라든가 스레드 같은 부분은 POSIX 표준으로 바로 작성할 수 있다. 미디어 계층에서는 C 언어와 Objective-C 언어를 섞어서 사용하는 프레임워크들이 제공된다. 3D 기능을 사용할 수 있는 OpenGL ES부터 이를 코코아에서 편하게 쓸 수 있도록 도와주는 코어 애니메이션 프레임워크, 그리고 소리 재생과 관련된 코어 오디오 프레임워크 등이 지원된다. 코코아 터치 계층은 아이폰 및 아이팟 터치용 API로서, 4개의 추상화 계층 중 가장 상위 계층에 존재하며, 개발자로 하여금 하위 계층을 자세히 알 수고를 덜어준다. GUI를 구현하고 이벤트-구동(event-driven) 기법을 쓰는 아이폰 및 아이팟 터치용 응용 소프트웨어는 보통 코코아 터치 계층에 기반하여 작성된다. 코코아 터치는 iOS에 대한 일종의 추상화 계층을 제공한다[16-20].

2.3 기존의 모바일 콘텐츠 변환기

그동안 국내 모바일 시장이 활성화 되었음에도 불구하고 모바일 콘텐츠 변환기에 대한 연구는 매우 부족하여 사례가 많지 않은데다가 기존의 모바일 콘텐츠 변환기는 대부분 동일한 프로그래밍 언어 환경에서의 변환만을 지원하거나 자동변환을 지원하지 않고 있어 프로그래머가 수작업으로 많은 부분을 변환시켜줘야 하는 실정이다.

기존의 모바일 콘텐츠 변환기 중에는 XML을 이용하여 자바 콘텐츠의 변환을 시도한 연구가 있었으며 [1-4], GVM 플랫폼의 모바일 C 콘텐츠를 WIPI C나 자바로 변환하는 연구가 있었다[6,7]. 또한, 변환 대상 소스 코드에서 사용되는 API를 실행 대상 환경에 거

의 유사한 형태로 동일 기능을 구현한 래퍼 함수 (Wrapper Function)를 정의하여 소스 코드의 변경 없이 변환 목적 플랫폼에서 동일한 동작을 하도록 BREW C와 WIPI C를 상호변환하거나[5], GVM C를 BREW C로 변환하는 연구가 있었지만[10] 변환할 때 소스코드가 자동으로 변환되지 않아 사용자가 개입하여 수동으로 변환을 해야 하는 등의 단점을 가지고 있다. 한편, 컴파일러 제작기술[8,9]을 활용하여 모바일 콘텐츠의 자동변환에 대한 연구가 시도되어 GVM 플랫폼의 모바일 C 콘텐츠를 WIPI C나 자바로 또는 MIDP 자바로 변환하는 연구[6,7]와 GNEX 콘텐츠를 위피 콘텐츠와 스마트폰 플랫폼인 안드로이드 콘텐츠로 변환하는 연구[11,12]가 발표되어 시간과 비용을 최소화하고 콘텐츠의 재사용을 높여 생산성을 향상시킬 수 있는 방법이 제시되었다. 또한, 최근에 스마트폰이 활성화됨에 따라 기존의 모바일 콘텐츠를 안드로이드나 아이폰과 같은 스마트폰 콘텐츠로 변환하는 변환[12] 작업이 일부 시도되고 있는 중이다.

3. GNEX-to-iPhone 변환기 시스템

모바일 콘텐츠 자동 변환기 시스템은 소스 형태의 콘텐츠를 입력 받아 다른 플랫폼에서 실행되는 콘텐츠의 소스 형태로 변환하는 소스 레벨 변환을 한다. 소스 레벨의 자동 변환을 위해서는 우선, 소스 코드가 변환 대상 플랫폼에서 동일한 동작을 수행하는 소스 코드로 변환되어야 하고, 이미지, 사운드 등의 리소스 데이터도 변환 대상 플랫폼에서 사용할 수 있는 형식으로 변환되어야 한다. 또한, 동일한 프로그래밍 환경, 이벤트 환경을 위한 API 라이브러리가 제공되어야 한다. 그림 3은 GNEX-to-iPhone 변환기 시스템의 구성도이다.

본 논문에서는 GNEX 콘텐츠를 iPhone OS 3.1

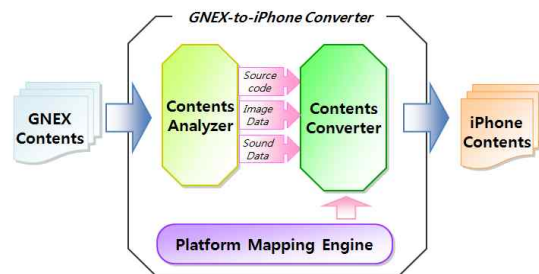


그림 3. GNEX-to-iPhone 변환기 시스템 구성도

표 1. 구현 환경

항 목	사용 환경
컴퓨터 사양	Pentium 4 3GHz 1GB RAM
운영체제	Windows XP Professional
개발 도구	Visual C++ .NET 7.0

Final 버전의 콘텐츠로 변환하는 GNEX-to-iPhone 변환기를 구현하였다. 구현 환경은 표 1과 같다.

3.1 콘텐츠 분석기

콘텐츠 분석기는 입력받은 GNEX 콘텐츠를 분석하여 모바일 C 소스 코드 형태로 변형된 리소스 데이터를 추출한다. 이미지, 사운드 타입의 변수 형태인 리소스 데이터를 각 변수별로 분리하여 콘텐츠 변환기에 전달한다. 그리고 리소스 부분을 제외한 나머지 소스 코드 부분도 콘텐츠 변환기에 전달한다. 다음 그림 4는 콘텐츠 분석기의 구조도이다.

GNEX에서는 파일 시스템을 지원하지 않기 때문에 리소스 데이터를 이진 파일(binary file) 형태로 사용하지 않고 문서 파일(text file) 형태로 소스 코드에 입력하여 사용한다. 따라서, 리소스 데이터의 변환을 위해서는 소스 코드에 입력되어 있는 리소스 데이터를 분리해야 한다[13-15]. 다음 그림 5는 GNEX의 리소스(이미지, 사운드) 데이터의 예제이다.

3.2 콘텐츠 변환기

콘텐츠 변환기는 콘텐츠 분석기로부터 전달받는 GNEX의 소스 코드, 이미지 데이터, 사운드 데이터를 변환 목적 플랫폼인 iPhone에서 사용할 수 있도록 변환해준다. 그림 6은 콘텐츠 변환기의 시스템 구성도를 나타낸 것이다.

GNEX-to-iPhone 변환기는 GNEX 콘텐츠를

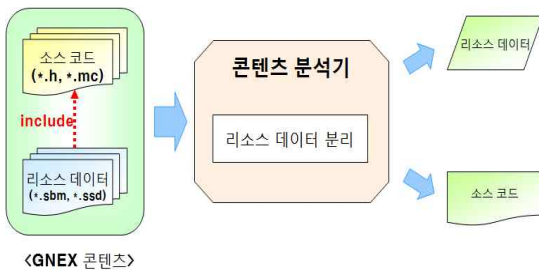


그림 4. 콘텐츠 분석기 구조도

```

30003 const sound sound2_2 =
30004 {
30005     0x06, 0x88,
30006     0x40, 0x40, 0x40, 0x44, 0x00, 0x00, 0x17, 0x87,
30007     0x43, 0x4E, 0x54, 0x49, 0x00, 0x00, 0x00, 0x05,
30008     0x00, 0x34, 0x01, 0xF0, 0x00, 0x4F, 0x50, 0x44,
30009     0x41, 0x00, 0x00, 0x00, 0x35, 0x44, 0x63, 0x68,
30010     0xFF, 0x00, 0x00, 0x00, 0x19, 0x45, 0x53, 0x00,
30011     0x01, 0xF8, 0x41, 0x30, 0x00, 0x06, 0x59, 0x76,
30012     0x31, 0x31, 0x43, 0x32, 0x41, 0x32, 0x00, 0x06,
3009 /*****
3010 const image img861[5] = {
3011     {
3012         0x00, 0x00, 0x40, 0x58, 0x00,
3013         0x00, 0x03,
3014         0x6F, 0x7A, 0x7E, 0x43, 0x6B, 0x7A, 0x25,
3015         0x65, 0x55, 0xA9, 0x55, 0x6A, 0xA5, 0x99,
3016         0x55, 0x6A, 0x55, 0x69, 0xA8, 0xA8,
3017         0x99, 0x55, 0x6A, 0x55, 0x5A, 0xA9, 0x66,
3018         0xA5, 0x95, 0x55, 0x66, 0x55, 0x96, 0xA8,
3019         0xA5, 0x95, 0x55, 0x55, 0x55, 0x65, 0x55,
    
```

그림 5. GNEX 콘텐츠의 리소스 데이터 예제

iPhone의 콘텐츠로 변환할 때 소스 코드를 iPhone 콘텐츠에서 그대로 사용하기 위해 필요하거나 불필요한 내용을 추가, 삭제해야 하는 부분이 있다. 콘텐츠 변환기의 소스 수정 모듈은 GNEX 콘텐츠의 소스 코드를 읽어가며 이와 같은 수정 작업을 수행한다. GNEX의 모바일 C 언어에서는 '%'와 '%' 기호 사이에 #DEFINE 매크로를 이용하여 응용 프로그램에 대한 정보를 정의하는데 이 정보는 변환 시 사용하지 않으므로 삭제한다.

GNEX에서는 VDI(Variable Depth Image)라는 자체 이미지 형식을 사용한다. 모바일 C의 변수 형태인 VDI 형식을 사용하여 다른 플랫폼에서는 제공하지 않는 감마 변환, 팔레트 변환, 역상 출력 등의 그래픽 기능들을 제공하고 있다[13-15]. 따라서, GNEX의 모든 그래픽 기능을 지원하기 위하여 이미지 형식은 VDI를 유지하고, iPhone에서 사용할 수 있도록 그래픽 기능을 지원하는 API를 제공하였다.

iPhone의 그래픽 API는 실제 LCD 버퍼에 출력하도록 되어있어서 사용 시 바로 화면에 출력이 된다. 플랫폼 매핑 엔진은 GNEX와 동일한 방식으로 그래픽 출력을 지원하기 위해 iPhone의 API를 이용해서 가상의 LCD 버퍼를 생성한다. 그리고 가상의 LCD 버퍼에 이미지, 도형, 문자를 출력하도록 GNEX의 그래픽 API와 동일한 형태의 그래픽 API를 구현하여 제공한다. Flush 메서드는 가상 LCD 버퍼의 내용을 실제 LCD 버퍼에 전달하도록 구현하여 GNEX와 동일한 방식으로 LCD에 출력하도록 하였다.

또한, GNEX에서는 두 개의 가상 LCD 버퍼를 사용하여 가상 LCD 버퍼의 내용을 부 가상 LCD 버퍼(Sub Virtual LCD Buffer)에 저장하였다가 다시 불

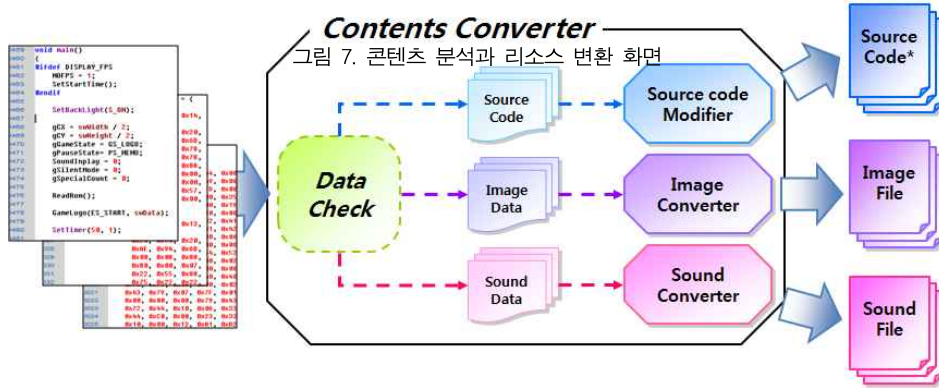


그림 6. 콘텐츠 변환기 시스템 구성도

러울 수 있는 기능을 제공하여 동일한 화면을 반복적으로 출력해야 하는 경우 효율적인 환경을 제공한다. 플랫폼 매핑 엔진에서는 이와 같은 기능을 제공하기 위해 부 가상 LCD 버퍼를 생성하고, 가상 LCD 버퍼와 데이터를 서로 주고받을 수 있는 API를 GNEX의 API와 동일한 형태로 제공한다. 다음 표 2는 GNEX VDI 형식의 이미지 타입 정보이다[13-15].

GNEX에서 사용하는 사운드 형식은 아마하의 MMF 형식이다. iPhone에서는 MMF 형식을 지원하지 않기 때문에 사운드 변환 모듈은 문서 파일 형식으로 되어있는 사운드 리소스 파일을 MMF 플레이어에서 재생 가능한 이진 파일 형식의 MMF 파일로 변환하여 출력하고 출력된 MMF 파일을 MP3 혹은 WAV 파일로 변환하는 프로그램을 시스템 명령어로 호출하여 수행한다.

그림 7은 GNEX 콘텐츠를 입력으로 받아 콘텐츠를 분석하여 소스 파일과 리소스 데이터를 분리하고 이

를 iPhone 콘텐츠의 리소스 파일로 변환한 화면이다. 터치 스크린을 지원하는 피쳐폰과 스마트폰이 등장하기 이전의 일반 피쳐폰의 콘텐츠들은 버튼을 이용해 조작하는 방식이었다. 따라서, 기존의 콘텐츠들은 모두 버튼을 누르면 이벤트가 발생하는 이벤트 방식으로 작성되어 있다. 그러나 많은 터치스크린을 지원하는 모바일 장치, 특히 iPhone의 경우는 단지 하나의 터치스크린과 홈 버튼으로 구성되어 있다. 이 경우는 콘텐츠가 상황에 따라 적재적소에 버튼 등을 화면에 표시하여 사용자의 터치를 받아서 이벤트를 발생시키는 방식이다. 따라서 화면 내에 콘텐츠의 실행 상태와 함께 사용되던 버튼들을 모두 표현하고 터치에 대한 키 입력 이벤트를 전달하여 기존 실행 환경과 유사한 환경을 제공하도록 하였다.

3.3 플랫폼 매핑 엔진

플랫폼 매핑 엔진(platform mapping engine)은

표 2. GNEX VDI 형식의 이미지 타입 정보

구분	Color Depth	Type	값	Bits Per Pixel	최대 사용 색상 수	Local Palette 크기(byte)
기본 VDI 형식	4Gray	GRAY1	0x02	1	2	1
		GRAY2	0x03	2	4	2
		GRAY4	0x04	4	16	0
	Color	COLOR1	0x05	1	2	2
		COLOR2	0x06	2	4	4
		COLOR4	0x07	4	16	16
		COLOR8	0x08	8	182	0
	True Color	TCOLOR1	0x09	1	2	최대6
		TCOLOR2	0x0A	2	4	최대12
		TCOLOR4	0x0B	4	16	최대48
TCOLOR8		0x0C	8	256	최대768	

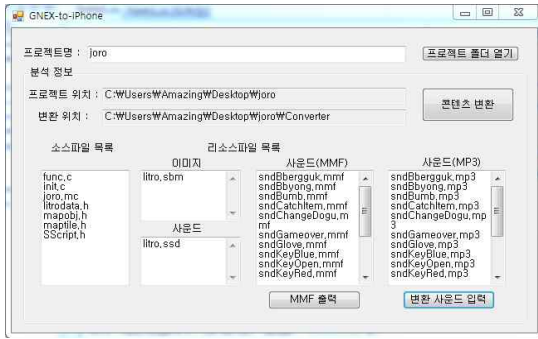


그림 7. 콘텐츠 분석과 리소스 변환 화면

GNEX 콘텐츠의 소스 코드에서 사용된 디스플레이, 그래픽, 사운드 출력, 시스템 변수, 이벤트 핸들러 등의 API를 변환 목적 스마트폰 플랫폼인 iPhone에서 사용할 수 있도록 변환하여 GNEX 콘텐츠가 iPhone 플랫폼에서 응용 프로그램으로 실행할 수 있게 해 주는 기능을 한다. 이를 위해 GNEX와 동일한 실행 환경을 구축하고 이를 바탕으로 GNEX의 API와 시스템 변수 등을 동일한 형태로 사용할 수 있도록 래퍼 함수(wrapper functions)를 사용하여 GNEX의 API를 iPhone 플랫폼의 API로 구현하였다.

특히, GNEX의 VDI 이미지 형식의 출력을 위한 환경, 함수를 만들었다. 이와 같이 함으로써 iPhone 콘텐츠로 번역된 소스 코드를 추가적으로 수정하지 않고 실행할 수 있도록 하였으며, GNEX와 동일한 형태의 API를 사용함으로써 번역된 소스 코드를 쉽게 이해하고 수정할 수 있게 하였다. 그림 8은 플랫폼

매핑엔진의 구조이고, 표 3과 4는 변환기 시스템에서 지원하는 시스템 변수의 목록과 API 목록이다.

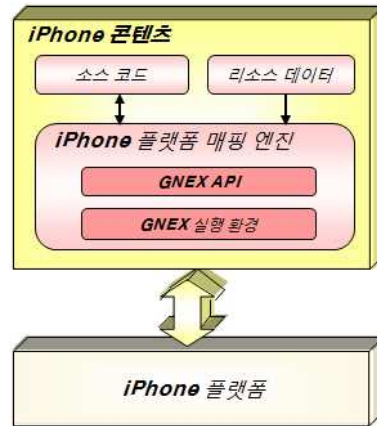


그림 8. 플랫폼 매핑 엔진의 구조도

표 3. 시스템 변수 목록

이름	내용
swData	이벤트 정보
swWidth	LCD 가로 크기
swHeight	LCD 세로 크기
swFrame	0~5039까지 반복되는 카운터
swFrame2	0~1까지 반복되는 카운터
swFrame3	0~2까지 반복되는 카운터
swFrame6	0~5까지 반복되는 카운터
swRsRcvBuf	시리얼 데이터 수신 버퍼

표 4. API 목록

구분	지원 함수
System(3)	GetDate, GetTime, Exit
Graphic(36)	SetClip, ResetClip, SetActiveBuffer, SetGamma, SetColor, SetFont, SetFontColor, SetFontAlign, SetFontType, SetPalette, SetImageAlpha, Clear, ClearRGB, ClearWhite, ClearBlack, PutPixel, CopyImage, CopyImageDir, CopyImagePal, CopyImageDirPal, CopyImageEx, CopyImageTile, DrawLine, DrawHLine, DrawRect, DrawStr, DrawText, DrawStrSolid, DrawStrSolid2, FillRect, FillRectEx, FillEllipse, SaveLCD, RestoreLCD, CopyLCD, Flush
Handset Control(14)	PlaySound, StopSound, SetVolume, StartVib, StopVib, SetBackLight, GetUserNV, PutUserNV, SetTimer, SetTimer1, SetTimer2, ResetTimer, ResetTimer1, ResetTimer2
String(15)	StrLen, StrCpy, StrSub, StrCat, GetChar, GetCharString, PutChar, AsciiToInt, PutByte, GetByte, PutBytes, GetBytes, MakeStr1, MakeStr2, MakeStr3
Mathematics(8)	Abs, Rand, RandRatio, Sin100, Cos100, ArrayToVar, ArrayToArray, HitCheck

4. 실행결과 및 분석

본 논문에서 구현한 GNEX-to-iPhone 변환기를 통해 GNEX 콘텐츠를 iPhone 콘텐츠로 변환한 후에 GNEX-to-iPhone 콘텐츠 변환기의 성능을 측정하고 분석하였다. 그림 9는 “Aiolos”, “무사도”, “Joro”의 3개의 게임을 GNEX 에뮬레이터와 iPhone 에뮬레이터에서 실행시켜 비교한 모습이다. 그래픽, 이미지 출력, 사운드 출력, 키 이벤트, 타이머 구동, 데이터 저장 등의 동작이 모두 동일하게 실행됨을 확인할 수 있었다.

콘텐츠 실행 속도는 초당 프레임 수(FPS: Frame Per Second)를 측정하여 비교하였다. 그림 10에서와 같이 변환된 iPhone 콘텐츠는 GNEX 콘텐츠와 비슷한 속도를 보여주거나 조금 빠른 속도를 보여주고 있어 GNEX-to-iPhone 변환기 시스템이 안정적임을 보여주고 있다.



그림 9. GNEX, iPhone 플랫폼에서의 콘텐츠 실행결과

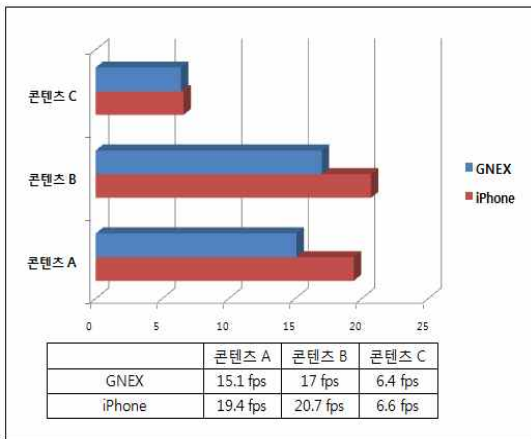


그림 10. 콘텐츠 실행속도비교(초당 프레임 수)

5. 결론 및 향후 연구

모바일 게임 콘텐츠 시장은 매년 높은 성장률의 시장 규모를 이어가며 모바일 시장 최고의 킬러 콘텐츠로 자리 잡았다. 하지만 모바일 플랫폼이 달라 하나의 모바일 게임 콘텐츠를 서비스하기 위하여 콘텐츠를 중복 개발하거나 변환해야하는 문제점이 있다. 본 논문에서 구현한 모바일 콘텐츠 자동 변환기인 GNEX-to-iPhone 변환기는 이런 문제를 해결할 수 있는 하나의 방법이다. 본 변환기는 기존의 모바일 콘텐츠인 GNEX 콘텐츠를 스마트폰 플랫폼인 iPhone 콘텐츠로의 변환 작업이 단기간 내에 자동으로 진행될 수 있도록 하여 GNEX 콘텐츠를 iPhone 플랫폼으로 이식하기 위한 개발 과정, 변환 과정에 중복 투자되던 시간 및 기간을 상당히 단축함으로써 생산성을 향상시킬 수 있도록 하였다.

앞으로 콘텐츠의 실행 속도를 높이기 위한 연구와 콘텐츠 변환기 시스템의 기능을 보완, 확장하여 기존의 콘텐츠들을 현재 확산되고 있는 안드로이드, 바다와 같은 다양한 스마트폰 플랫폼에서 실행될 수 있도록 변환기를 확장할 예정이다.

참고 문헌

- [1] 김미영, 무선 인터넷 서비스를 위한 XML 기반 콘텐츠 변환 시스템의 설계 및 구현, 영남대학교 석사학위논문, 2003.
- [2] 김석훈, J2ME MIDP를 이용한 XML 기반의 모바일 콘텐츠 변환 시스템 설계 및 구현, 한남대학교 석사학위논문, 2003.
- [3] 김영선, 장덕철, “XML Parser 추출에 의한 모바일 콘텐츠 변환 설계,” 멀티미디어학회 논문지, Vol.6, No.2, pp. 267-274, 2003.
- [4] 윤성일, Mobile 기반의 유무선 플랫폼 통합 변환 시스템, 한남대학교 박사학위논문, 2003.
- [5] 이영중, WIPI와 BREW 플랫폼 간 C 언어 기반 솔루션 변환 방법, 충남대학교 석사학위논문, 2007.
- [6] 박상훈, 권혁주, 김영근, 이양선, “모바일 콘텐츠의 자동변환 위한 GVM-to-MIDP 변환기의 설계 및 구현,” 한국멀티미디어학회 학회지, Vol.9, No.2, pp. 215-218, 2006.

[7] 박상훈, 권혁주, 김영근, 이양선, “모바일 게임 콘텐츠를 위한 GVM-to-WIPI 자동 변환기의 설계 및 구현,” 한국정보처리학회 게임 논문지, 제3권, 제1,2호, pp. 51-60, 2006.

[8] 이양선, 황대훈, 나승원, “JVM 플랫폼에서 .NET 프로그램을 실행하기 위한 MSIL-to-Bytecode 번역기의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.7, No.7, pp. 976-984, 2004.

[9] 이양선, 나승원, 황대훈, “Intermediate Language Translator for Execution of Java Programs in .NET Platform,” 한국멀티미디어학회 논문지, Vol.7, No.6, pp. 824-831, 2004.

[10] 이양선 외 4인, “모바일 게임 콘텐츠의 자동 변환을 위한 GVM-to-BREW 번역기 시스템,” 한국정보처리학회 게임 논문지, Vol.2, No.1, pp. 49-64, 2005.

[11] 이양선, “모바일 콘텐츠의 자동 변환을 위한 GNEX C-to-WIPI Java 변환기의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.13, No.4, pp. 609-617, 2010.

[12] 손윤식, 오세만, 이양선, “소스 레벨 콘텐츠 변환기를 이용한 GNEX C-to-Android Java 변환기의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.13, No.7, pp. 1051-1061, 2010.

[13] 신지소프트, GNEX SDK, http://www.gnexclub.com/download/download_a1.jsp

[14] 신지소프트, Mobile C Library Function Reference, <http://www.gnexclub.com/>

[15] 신지소프트, Mobile C Programming Guide, <http://www.gnexclub.com/>

[16] 라떼, 아이폰 4 아이패드 완전정복, 정보문화사, 2010.

[17] Backlin, Eugene, *Professional iPhone and iPad SDK Application Development*, Wrox, 2010.

[18] Apple, iOS Reference Library, iOS Technology Overview, <http://developer.apple.com/devcenter/ios>

[19] Apple, iOS Reference Library, Cocoa Fundamentals Guide, <http://developer.apple.com/devcenter/ios>

[20] <http://developer.apple.com/devcenter/ios>



이 양 선

1985년 동국대학교 컴퓨터공학과 공학사
 1987년 동국대학교 대학원 컴퓨터공학과 공학석사
 1993년 동국대학교 대학원 컴퓨터공학과 공학박사

1994년 3월~현재 서경대학교 컴퓨터공학과 교수
 1996년 3월~2000년 2월 서경 대학교 전자계산소 소장
 2000년 2월~현재 한국멀티미디어학회 이사
 2005년 1월~2006년12월 한국멀티미디어학회 총무이사
 2006년 1월~현재 한국정보처리학회 이사, 게임연구회 위원장
 2009년 1월~2009년 12월 한국멀티미디어학회 부회장, 논문지 편집위원장
 2011년 3월~현재 (사)스마트개발자협회 이사
 관심분야: 스마트폰 솔루션, 임베디드 SW, 게임평가 모델, CT 기술 등



최 효 준

2009년 서경대학교 컴퓨터공학과 공학사
 2011년 서경대학교 대학원 컴퓨터공학과 대학원생
 관심분야: 스마트폰 솔루션, 임베디드 SW 등



김 재 성

2010년 서경대학교 컴퓨터공학과 공학사
 2011년 서경대학교 대학원 컴퓨터공학과 대학원생
 관심분야: 스마트폰 솔루션, 임베디드 SW 등