

모바일 콘텐츠 자동변환기를 위한 콘텐츠 분석기와 리소스 변환기 개발

이양선[†], 김재성^{**}, 김명진^{***}

요 약

국내 이동통신사들이 서로 다른 모바일 플랫폼을 채택하여 사용함으로써 인해 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 기존의 WIPI나 GNEX와 같은 모바일 플랫폼의 게임 콘텐츠를 Android나 iOS와 같은 스마트 플랫폼의 콘텐츠로 자동 변환해주는 모바일 콘텐츠 자동 변환기 시스템의 콘텐츠 분석기와 리소스 변환기를 구현하였다. 모바일 콘텐츠 자동 변환기 시스템은 단시간 내에 스마트 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

Yang-Sun Lee[†], JaeSung Kim^{**}, MyoungJin 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 contents analyzer and the resource converter of the automatic mobile contents converter system that converts game contents of the mobile platform such as the WIPI and the GNEX to contents of the smart platform such as the Android and the iOS. The automatic mobile contents converter system helps minimize the amount of time and cost required in servicing contents to different mobile communication companies by promptly converting platform-specific-contents to run under smart platforms. Also, the automatic conversion 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 game contents.

Key words: Automatic Mobile Contents Converter(모바일 콘텐츠 자동 변환기), Mobile Platform(모바일 플랫폼), Smart Platform(스마트 플랫폼), Contents Analyzer(콘텐츠 분석기), Resource Converter(리소스 변환기), Android(안드로이드), iPhone(아이폰), WIPI(위피), GNEX(지넥스)

※ 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 4월 22일, 수정일: 2011년 5월 12일
완료일: 2011년 5월 12일

[†] 종신회원, 서경대학교 컴퓨터공학과 교수

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

^{***} 정회원, 서경대학교 컴퓨터공학과 대학원생
(E-mail: highkrs@naver.com)

※ 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 다른 플랫폼에서 서비스하기 위해 시간과 비용이 중복 투자되고 있다 [1-5].

본 논문에서는 이런 문제를 해결하기 위해 기존의 피쳐폰 플랫폼의 콘텐츠를 스마트폰 플랫폼의 콘텐츠로 자동으로 변환해주는 모바일 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[6-13].

본 논문에서는 모바일 콘텐츠 자동 변환기 시스템의 콘텐츠 분석기와 리소스 변환기를 개발하였다. 피쳐폰의 WIPI나 GNEX 플랫폼에서 사용되던 기존의 모바일 게임 콘텐츠를 스마트 플랫폼인 iOS(iPhone)이나 Android 콘텐츠로 자동 변환할 수 있도록 함으로써 기존 콘텐츠의 재사용성을 높이고 스마트폰 사용자가 보다 다양한 콘텐츠를 제공받을 수 있도록 하였다.

2. 관련 연구

2.1 WIPI 플랫폼

WIPI(Wireless Internet Platform for Interoperability)는 한국 무선 인터넷 표준화 포럼(KWISF)에 의해 제정되고, 한국정보통신기술협회(TTA)에 의해 TTAS.KO-60.0036으로 채택된 이동통신 단말기용 응용프로그램 실행 환경을 표준화한 규격이다. 이동통신 사업자들이 각기 다른 플랫폼을 사용함으로써 단말기 제조사와 콘텐츠 업체들의 개발 부담이

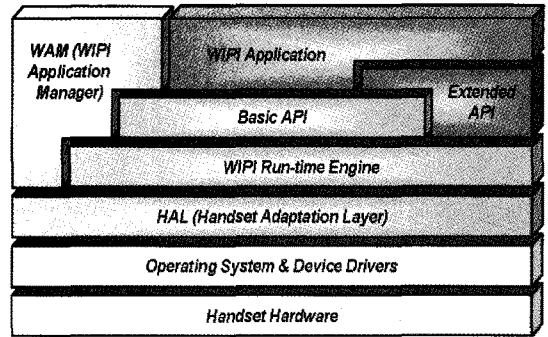


그림 1. WIPI 플랫폼의 구조

크다는 점 등을 들어 국내 무선인터넷 플랫폼 표준화의 필요성이 대두되면서 제정한 국내 표준 규격이다. 그림 1은 WIPI 플랫폼의 구조를 나타낸 것이다.

WIPI는 응용프로그램 개발 언어로 네이티브(Native) 방식의 C와 Java를 모두 지원한다. Java의 경우 AOTC(Ahead Of Time Compiler)를 통해 C로 변환하여 다시 네이티브 방식으로 실행된다. 위피 규격은 크게 HAL(Handset Adaptation Layer)과 기본 API(Application Programming Interface)로 구성된다. HAL은 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층이다. 기본 API는 표준화된 플랫폼 호환성을 제공해 다양한 응용 프로그램 개발을 촉진하기 위한 기본 API 모음으로 C와 자바 API로 구성되어 있다[14,15].

2.2 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 핸들러(Event Handler)는 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 플랫폼에서 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

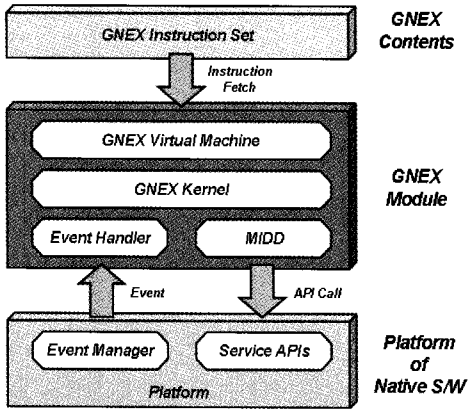


그림 2. GNE X의 구조

Depth Image)라는 단말기 전용으로 설계된 이미지 형식을 사용한다. VDI는 픽셀(Pixel) 당 할당되는 비트 수를 가변적으로 정의하여 사용하는 일종의 비트 맵 형식의 이미지 규격이다. GNE X 응용 프로그램에서는 BMP와 같은 이미지 파일을 GNE X SDK에서 제공하는 이미지 도구를 통해 VDI 형식으로 변환하여 모바일 C 소스 코드에 포함시킨다. 사운드 리소스 역시 GNE X 응용 프로그램에서 사용할 수 있도록 GNE X 규격의 사운드 파일로 변환하여 모바일 C 소스 코드에 포함하여 사용한다[16-18].

2.3 iOS 플랫폼

iPhone 플랫폼인 iOS(iPhone Operating System)는 애플의 스마트폰인 아이폰(iPhone)과 디지털 미디어 재생기인 아이팟 터치(iPod touch), 태블릿형 컴퓨터인 아이패드(iPad)에 내장되어 있는 운영체제이다. 실제로는 Mac OS X을 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 구성되어있다. 그림 3은 iOS의 구조를 나타낸 것이다. 코어 OS와 코어 서비스 계층은 기본적인 시스템 API등을 제공한다.

iOS는 Mac OS X와 같은 유닉스 기반으로서 POSIX 표준을 준수하는 API들을 제공한다. 네트워

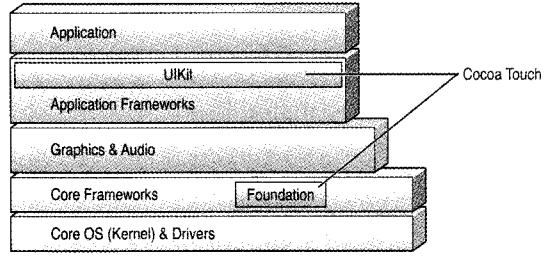


그림 3. iOS 플랫폼의 구조

크 소켓에 관련된 부분이라든가 스탠드 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[19-22].

2.4 Android 플랫폼

안드로이드(Android) 플랫폼은 구글에서 개발한 운영체제, 미들웨어, 응용 프로그램을 포함하는 모바일 기기에 최적화된 플랫폼이다. 안드로이드 플랫폼은 오픈 소스 정책을 채택하였으며, 리눅스 커널, 라이브러리, 런타임, 어플리케이션 프레임워크, 어플리케이션으로 구성되어 있다.

그림 4는 안드로이드 플랫폼의 계층 구조와 구성요소를 도시한 것이다. 리눅스 커널은 보안, 메모리 관리, 프로세스 관리, 네트워크 스택, 드라이버 모델과 같은 리눅스 버전 2.6의 핵심 시스템 서비스를 이용하며, 커널은 하드웨어와 소프트웨어 간의 추상 계층으로 동작한다. 라이브러리는 C와 C++로 구성되었으며 C 시스템 라이브러리, 미디어 라이브러리, 3D 라이브러리 등을 제공한다. 어플리케이션 프레임워크는 자바로 구성된 패키지 컴포넌트이며, 어플리케이션은 이 프레임워크의 패키지를 사용하여 작성

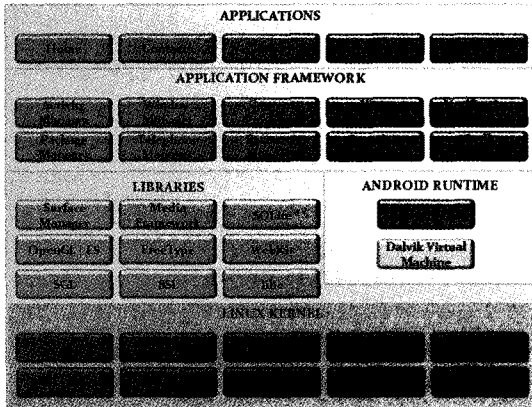


그림 4. 안드로이드 플랫폼의 계층 구조

할 수 있다.

안드로이드 플랫폼의 모든 응용 프로그램은 자바로 작성되며, 작성된 응용 프로그램은 자바 컴파일러에 의해 클래스 파일로 변환된다. 생성된 클래스 파일은 실행에 앞서 DEX(Dalvik Executable File) 형식으로 다시 변환되며, dalvik 가상기계체에 의해 실행된다. DEX 파일은 저장 공간과 메모리 사이의 효과적인 매핑 정의에 최적화된 포맷이며, dalvik 가상기계체는 레지스터 기반의 가상기계로 제한된 메모리에 최적화된 특징을 가진다[23-25].

2.5 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.

먼저, 기존의 모바일 콘텐츠 변환기 중에는 XML을 이용하여 Java 콘텐츠의 변환을 시도한 연구가 있었으며[1-4], GVM 플랫폼의 모바일 C 콘텐츠를 WIPI C나 Java로 변환하는 연구가 있었다[6,7]. 또한, 변환 대상 소스 코드에서 사용되는 API를 실행 대상 환경에 거의 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 제작기술을[8,9] 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 위퍼 콘텐츠와 스마트폰 플랫폼인 안드로이드나 아이폰 콘텐츠로 변환하는 연구[11-13]가 발표되어 시간과 비용을 최소화하고 콘텐츠의 재사용을 높여 생산성을 향상시킬 수 있는 방법이 제시되었다. 또한, 최근에 스마트폰이 활성화됨에 따라 기존의 피쳐폰 모바일 콘텐츠를 안드로이드나 아이폰과 같은 스마트폰 콘텐츠로 변환[12,13]하거나 스마트폰 콘텐츠를 다른 스마트폰 콘텐츠로 변환하는 작업을 시도하려고 준비하고 있는 중이다.

3. 콘텐츠 분석기와 리소스 변환기

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 5는 모바일 콘텐츠 자동 변환기 시스템 모델을 나타낸 것이다.

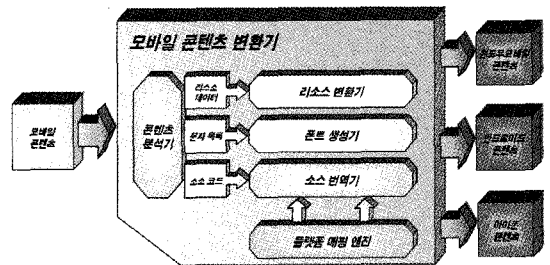


그림 5. 스마트폰을 위한 모바일 콘텐츠 자동 변환기 시스템 모델

3.1 콘텐츠 분석기

콘텐츠 분석기(content analyzer)는 입력받은 콘

텐츠를 분석하여 콘텐츠에 내재된 리소스 데이터와 소스 코드를 분리하여 출력하는 시스템이다. 기존의 GNEX와 같은 플랫폼은 파일 시스템을 지원하지 않고 리소스 데이터와 사용자 데이터를 텍스트 형태로 소스 코드에 입력하여 사용한다. 따라서, 변환에 앞서 소스 코드와 리소스 데이터를 분리하여 각각 변환해야만 한다. 즉, 콘텐츠 분석기는 이미지나 사운드 타입의 변수 형태인 리소스 데이터를 각 변수별로 분리하고 구조체에 정보를 담아 리소스 목록을 생성하고 실제 이미지나 사운드 리소스 데이터 인지, 맵 타일(map tile) 등의 사용자 데이터 인지 구분하여 리소스 변환기에 전달한다. 또한, 콘텐츠에 사용된 문자를 추출하여 문자 목록을 만들어 목적 플랫폼에서 동일한 문자를 사용할 수 있도록 폰트를 생성할 수 있게 한다. 그리고 리소스 부분을 제외한 나머지 소스 코드 부분은 소스 번역기에 전달하여 목적 플랫폼의 소스 언어로 자동번역하게 한다.

3.1.1 WIPI Java 콘텐츠 분석기

WIPI Java 콘텐츠 분석기는 WIPI Java 콘텐츠를 입력으로 받아 변환하려는 목적 플랫폼의 콘텐츠로 변환하기 쉽도록 분석하여, 소스 코드 파일, 리소스 파일 그리고 그 밖의 파일로 나누어 주게 된다. WIPI Java 콘텐츠 분석기의 구성은 WIPI Java 콘텐츠에서 리소스와 소스 코드, 기타 파일을 구분해주는 콘텐츠 분석 단계, 소스 파일을 분류하여 목록을 작성해주는 소스 파일 분류 단계, 이미지 파일을 분류하여 목록을 작성해주는 이미지 파일 분류 단계, 사운드 파일을 분류하여 목록을 작성해 주는 사운드 파일 분류 단계로 구성되어 있다. 그림 6은 WIPI Java 콘텐츠 분석기의 시스템 모델이다.

콘텐츠 분석 단계는 변환에 필요한 파일만을 추려내서 분류한다. 그리고 변환에 필요한 파일들 중에서

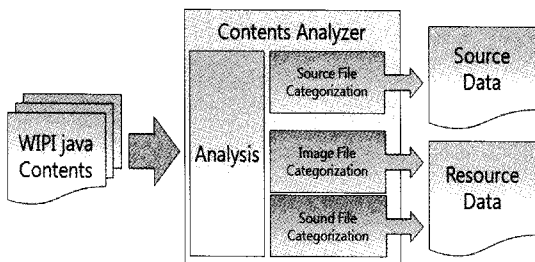


그림 6. WIPI Java 콘텐츠 분석기 시스템 모델

.java 확장자를 가진 파일을 분류하여 목록을 작성해주는 단계가 소스 파일 분류 단계이다. 또는 .png나 .bmp 등 WIPI Java에서 사용되는 이미지 포맷을 분류하여 목록을 작성해주는 단계가 이미지 파일 분류 단계이다. 사운드 분류 단계도 마찬가지로 .mmf나 .mid 등 WIPI Java에서 사용되는 사운드 포맷을 분류하여 목록을 작성해준다. 이미지 목록과 사운드 목록을 합쳐서 리소스 데이터를 작성해 주게 된다.

표 1과 표 2는 WIPI Java에서 지원하는 이미지 포맷 타입과 사운드 포맷의 타입을 나타낸 것이다.

표 1. WIPI Java 지원 이미지 파일 타입

타입	데이터 포맷
BMP	Bitmap
GIF	Graphic Interchange Format
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
SIS	Simple Image Service

표 2. WIPI Java 지원 사운드 파일 타입

타입	데이터 포맷
Qualcomm_CMX	Qualcomm CMX
Yamaha_MA1	Yamaha MA1
Yamaha_MA2	Yamaha MA2
Yamaha_MA3	Yamaha MA3 Single Channel Format
Yamaha_MA5	Yamaha MA3
Yamaha_SMAF	Yamaha Single Channel Format
Yamaha_SMAF- Phrase	Yamaha Multi Channel Format
Yamaha_SMAF- Audio	Yamaha SMAF-Audio Format
audio/ONEPOLY	One Poly Media Format
audio/GVMONEPOLY	GVM One Poly Media Format
audio/MIDI	MIDI
audio/MP3	MP3
audio/TONE	Tone
audio/FREQTONE	Frequency Tone
IS96	QCELP-8K
IS96A	QCELP-8K
IS733	QCELP-13K
IS127	EVRC-8K
G.723.1	G.723.1
audio/AAC	AAC
audio/AAC+	AAC+
AMR-WB	WCDMA용 음성 코덱
AMR-NB	

콘텐츠 분석기는 WIPI Java에서 지원하는 여러 가지 파일 포맷을 고려하여 이미지 분류 단계와 사운드 분류 단계에서 각각 위와 같은 포맷으로 분류하여 목록을 작성하여 준다. 그림 7과 그림 8은 콘텐츠 분석기의 실행결과를 나타낸 것이다. 콘텐츠 분석기는 WIPI Java 콘텐츠를 입력으로 받아 콘텐츠 목록을 작성하고 동시에 원본 파일에 변화를 주지 않기 위하여 각각을 분류하여 원본 콘텐츠의 폴더 내에 Converter라는 폴더를 생성하여 그 안에 모든 파일들을 복사하여 분류한다.

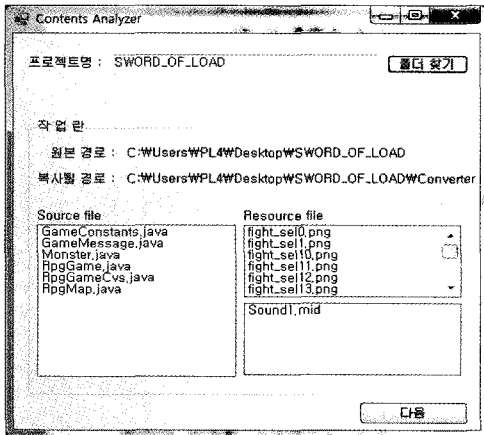


그림 7. WIPI Java 콘텐츠 분석기의 실행결과

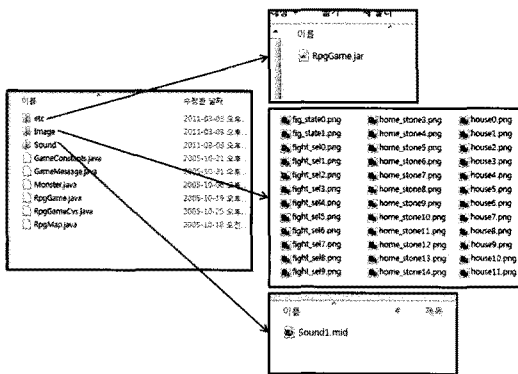


그림 8. WIPI Java 콘텐츠의 분석 결과-소스와 리소스

3.1.2 GNEX 콘텐츠 분석기

GNEX 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 콘텐

츠 변환기에 전달한다. 그리고 리소스 부분을 제외한 나머지 소스 코드 부분도 콘텐츠 변환기에 전달한다. 다음 그림 9는 GNEX 콘텐츠 분석기의 구조도이고, 그림 10은 GNEX 콘텐츠 분석기의 실행결과를 나타낸 것이다.

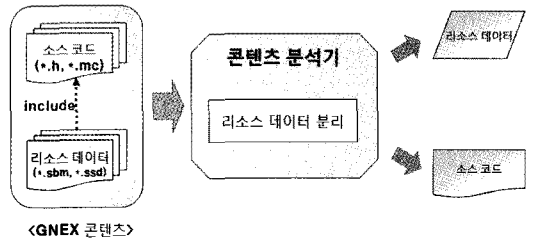


그림 9. GNEX 콘텐츠 분석기 구조도

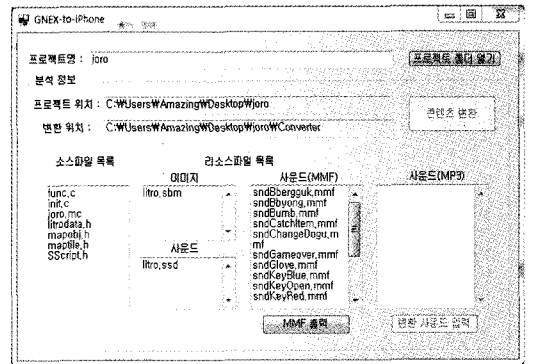


그림 10. GNEX 콘텐츠 분석기의 실행결과

3.2 리소스 변환기

리소스 변환기(resource converter)는 입력 콘텐츠에 사용된 텍스트 또는 바이너리 형태의 리소스 데이터를 목적 플랫폼의 파일 시스템에서 사용할 수 있도록 이미지 데이터와 사운드 데이터, 그리고 사용자 데이터로 변환하여 각 파일로 변환하는 시스템이다. 이를 위해 각 플랫폼에서 사용하는 이미지 파일 형식(차체 이미지 형식-VDI, BMP, GIF 등), 사운드 형식(MMF 등), 사용자 데이터를 조사하고 목적 플랫폼에서 사용할 수 있도록 변환해 주어야 한다.

즉, 기존의 모바일 플랫폼에서 이미지 파일 형식으로 지원하는 BMP나 PNG, JPEG 등과 사운드 형식으로 지원하는 야마하 형식(ma1, ma2,...) wav, mp3 등은 스마트폰 플랫폼인 안드로이드나 아이폰에서 지원하고 있기 때문에 변환 없이 텍스트 형태에서 바이너리 파일이나 사운드 파일 형태로 출력하면

된다. 그리고 어떤 리소스가 사용되었는지를 나타내는 관리파일만 작성해주면 된다. 그리고 자체 형식을 사용한 경우에는 그 기능을 지원하기 위해 형식은 유지하고 텍스트 형태에서 해당 플랫폼의 형식으로 변환하고 해당기능을 지원하는 API 함수를 통해 사용할 수 있게 해주면 된다. 또한, 텍스트 형태의 사용자 데이터도 해당 플랫폼의 파일 시스템을 통해 접근 가능하도록 바이너리 파일 형태로 출력해야 한다.

3.2.1 WIPI Java-to-Android Java 리소스 변환기

WIPI Java-to-Android Java 리소스 변환기는 콘텐츠 분석기가 WIPI Java 콘텐츠를 입력으로 받아 분석하여 출력한 파일 중 리소스 파일을 입력으로 받아, 즉 이미지 파일과 사운드 파일을 입력으로 받아 변환하려는 목적 플랫폼인 Android Java에서 지원하는 리소스 파일 포맷 형식으로 변환하는 시스템이다. 그림 11은 리소스 변환기 시스템 모델을 나타낸 것이다.

리소스 변환기는 WIPI Java의 리소스 파일을 입력으로 받아 이 리소스가 Android에서 사용 가능한 리소스인지 판단을 하여 만약 Android에서 사용할 수 없는 리소스 파일 포맷일 경우 그 리소스들을 각 변환기를 통해 Android에서도 사용이 가능하도록 변환을 한다. WIPI Java에서 사용되는 리소스 포맷 형식 중에서는 대부분 Android에서 호환이 되기 때문에 그대로 사용이 가능하지만 사운드 형식 중 Yamaha의 MMF 형식의 포맷은 Android에서 사용이 불가능하다. 따라서 Android에서 사용이 가능하도록 변환해야 한다.

리소스 변환기에서 하는 또 하나의 일은 Android의 프로젝트를 생성해준다. 변환 후 Android SDK에서 바로 접근을 하게 하기 위하여 Android 프로젝트에 관한 모든 파일을 생성한 후에 변환한 리소스들을 그 프로젝트에 적용하게 된다. 이와 같이 하면 별도

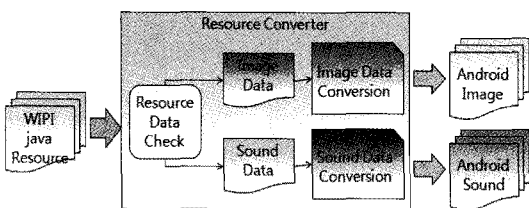


그림 11. WIPI Java-to-Android Java 리소스 변환기 시스템 모델

의 프로젝트 생성 할 필요가 없기 때문에 번거로움을 피할 수 있다. Android에서는 리소스 파일명이 제한되어있다. Android에서는 (“a-z” “0-9” “_” “.”)만 사용할 수 있다. 따라서 WIPI Java에서 사용되는 리소스 중에서 대문자를 소문자로 바꾸어 주어야 하는 경우가 있다. 리소스 변환기에서는 이와 같이 리소스들의 파일명을 하나하나 체크하여, 대문자를 소문자로, 그리고 특수 문자가 있을 경우에는 다른 대체 문자로 변환하여 준다. 그림 12와 그림 13은 리소스 변환기의 실행결과를 나타낸 것이다.

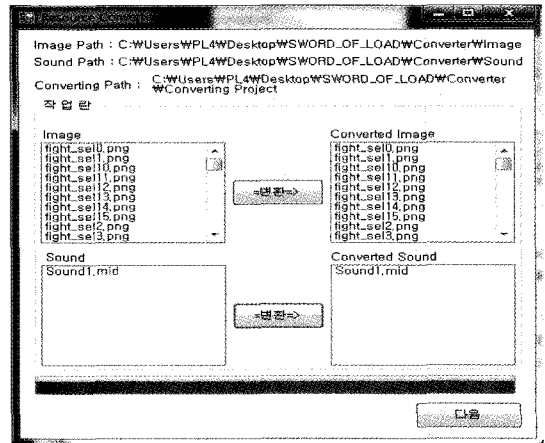


그림 12. WIPI Java-to-Android Java 리소스 변환기의 실행결과

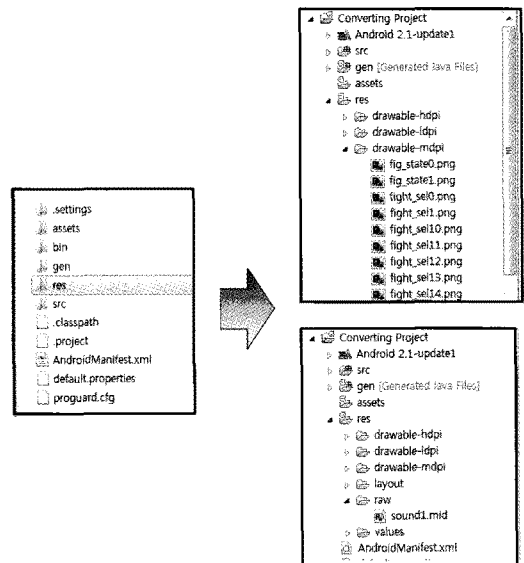


그림 13. WIPI Java-to-Android Java 리소스 변환기에서 생성된 Android 프로젝트와 리소스들

3.2.2 GNEX-to-iPhone 리소스 변환기

GNEX-to-iPhone 리소스 변환기는 콘텐츠 분석기로부터 전달받은 GNEX의 이미지 데이터, 사운드 데이터를 변환 목적 플랫폼인 iPhone에서 사용할 수 있도록 변환해준다. 그림 14는 리소스 변환기를 포함한 GNEX 콘텐츠 변환기 시스템 모델을 나타낸 것이다.

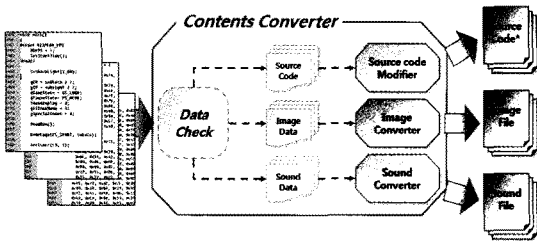


그림 14. GNEX-to-iPhone 콘텐츠 변환기 시스템 구성도

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[13-16]. 그림 15는 GNEX의 리소스(이미지, 사운드) 데이터의 예제이다.

```

30003 const sound sound2_2 =
30004 {
30005     0x06, 0x00,
30006     0x4D, 0x4D, 0x4D, 0x44, 0x00, 0x00, 0x17, 0x07,
30007     0x42, 0x4E, 0x54, 0x49, 0x00, 0x00, 0x00, 0x05,
30008     0x00, 0x34, 0x01, 0xFD, 0x00, 0x0F, 0x50, 0x44,
30009     0x41, 0x00, 0x00, 0x00, 0x25, 0x44, 0x63, 0x68,
30010     0xFF, 0x00, 0x00, 0x00, 0x19, 0x45, 0x50, 0x00,
30011     0x01, 0xF8, 0x41, 0x30, 0x00, 0x06, 0x50, 0x76,
30012     0x21, 0x31, 0x43, 0x32, 0x41, 0x32, 0x00, 0x06,
30013 //*****
30014 const image img061[5] = {
30015     {
30016         0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
30017         0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
30018         0x0F, 0x7A, 0x7E, 0x43, 0x6B, 0x7A, 0x25,
30019         0x65, 0x55, 0xA9, 0x55, 0x6A, 0xA5, 0x99,
30020         0x55, 0x6A, 0x55, 0x69, 0xA0, 0x96, 0xA0,
30021         0x99, 0x55, 0x6A, 0x55, 0x50, 0xA9, 0x66,
30022         0xA5, 0x95, 0x55, 0x66, 0x55, 0x96, 0xA0,
30023         0xA5, 0x95, 0x55, 0x65, 0x55, 0x55,
    
```

그림 15. GNEX 콘텐츠의 리소스 데이터

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, 영상 출력 등의 그래픽 기능들을 제공하고 있다[13-16]. VDI 형식을 iPhone에서 사용할 수 있는 그래픽 형식으로 바꾸면 이러한 기능들을 지원하기 힘들게 된다. 따라서, GNEX

의 모든 그래픽 기능을 지원하기 위하여 이미지 형식은 VDI를 유지하고, iPhone에서 사용할 수 있도록 그래픽 기능을 지원하는 API를 제공하는 방법을 선택하였다.

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[13-16].

GNEX에서 사용하는 사운드 형식은 야마하의 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 16은 GNEX 콘텐츠를 입력으로 받아 콘텐츠를 분석하여 소스 파일과 리소스 파일로 변환한 화면이다.

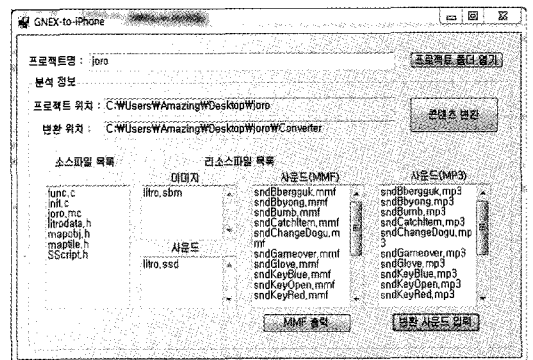


그림 16. GNEX 콘텐츠 분석과 GNEX-to-iPhone 리소스 변환 화면

4. 실행결과 및 분석

본 논문에서 구현한 모바일 콘텐츠 자동 변환기 시스템의 콘텐츠 분석기와 리소스 변환기를 적용하

여 피쳐폰의 WIPI나 GNEX 플랫폼의 모바일 게임 콘텐츠를 스마트 플랫폼인 iPhone이나 Android 콘텐츠로 자동 변환하여 그 결과를 비교하여 보았다. 변환기는 WIPI Java-to-Android Java 변환기, GNEX-to-iPhone 변환기를 통해 “Aiolos”, “무사도”, “Elemental Force”, “Zoro” 등의 게임을 각 에뮬레이터에서 실행시켜 비교한 모습이다. 그래픽, 이미지 출력, 사운드 출력 등의 동작이 모두 동일하게 실행됨을 확인할 수 있었다.

1) WIPI Java-to-Android Java 변환기 실행결과

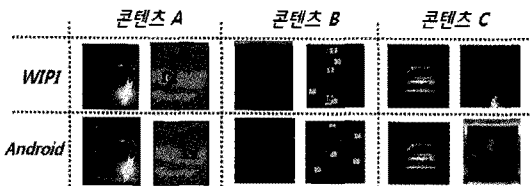


그림 17. WIPI와 Android 플랫폼에서의 콘텐츠 실행결과

2) GNEX-to-iPhone 변환기 실행결과

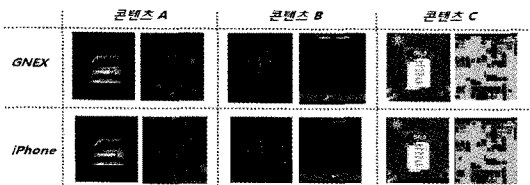


그림 18. GNEX와 iPhone 플랫폼에서의 콘텐츠 실행결과

5. 결론 및 향후 연구

본 논문에서 구현하려는 모바일 콘텐츠 자동 변환기는 기존의 모바일 콘텐츠를 스마트폰 플랫폼의 콘텐츠로의 변환 작업이 단기간 내에 자동으로 진행될 수 있도록 하여 피쳐폰 플랫폼의 콘텐츠를 스마트폰 플랫폼으로 이식하기 위한 개발 과정과 변환 과정에서 중복 투자되던 시간 및 기간을 상당히 단축함으로써 생산성을 향상시킬 수 있다.

본 논문에서는 모바일 콘텐츠 자동 변환기 시스템의 콘텐츠 분석기와 리소스 변환기를 개발하였다. 피쳐폰의 WIPI나 GNEX 플랫폼에서 사용되던 기존의 모바일 게임 콘텐츠를 스마트 플랫폼인 iPhone이나 Android, Windows Phone 7 콘텐츠로 자동 변환할

수 있도록 함으로써 기존 콘텐츠의 재사용성을 높이고 스마트폰 사용자가 보다 다양한 콘텐츠를 제공받을 수 있도록 하였다.

앞으로 모바일 콘텐츠 자동 변환기 시스템의 플랫폼 매핑 엔진과 소스 코드 자동번역기를 구현하여 전체 시스템을 완성할 예정이다. 또한, 구현과정에서 콘텐츠의 실행 속도를 높이기 위한 연구와 콘텐츠 변환기 시스템의 기능을 보완, 확장하여 기존의 콘텐츠들을 현재 확산되고 있는 Android, iOS(iPhone), Windows Phone 7, bada와 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 C-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 자동변환을 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] 이양선, 김재성, 최효준, "스마트폰 게임 콘텐츠를 위한 GNEX-to-iPhone 변환기의 설계 및 구현," 한국멀티미디어학회 논문지, Vol.14, No.4, 2011.

[13] 강진영, 정찬성, WIPI GNEX를 이용한 모바일 프로그래밍, 생능출판사, 2006.

[14] 한국 무선인터넷 표준화 포럼, 모바일 표준 플랫폼 규격, <http://www.kwisforum.org/>

[15] 박건태, 김승엽, Jlet으로 배우는 위피 프로그래밍, 한빛미디어, 2005.

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

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

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

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

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

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

[22] David Mark, Jack Nutting and Jeff LaMarche,

Beginning iPhone 4 Development : Exploring the iOS SDK, Apress, 2011.

[23] Goole, Android, <http://code.google.com/intl/ko/android/>

[24] J. Steele and N.To, *The Android Developer's Cookbook*, Addison-Wesley, 2010.

[25] S. Hashimi and S.Komatineni, D.MacLean, *Pro Android 3*, Apress, 2011.



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 게임연구회 위원장
 2006년 1월~현재 한국정보처리학회 이사
 2009년 1월~2009년 12월 한국멀티미디어학회 부회장, 논문지 편집위원장
 2011년 3월~현재 (사)스마트개발자협회 이사
 관심분야 : 스마트 시스템 솔루션, 임베디드 SW, 게임 평가모델, CT 기술 등



김 재 성

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



김 명 진

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