

모바일 프로그래밍 환경분석 및 구현

백현옥*, 조태경*, 이병효**, 이복용**
 상명대학교 정보통신과*, 동서울대학 전기정보제어과**

A Study on the Environment of Mobile Programing and Application

Hyun-Ok Pack*, Tae-Kyung Cho*, Byung-Hyo Lee**, Bok-Yong Lee**
 Dept of Information Telecommunication, Sangmyung University*, DONG SEOULI College**

Abstract - 본 논문에서는 모바일 프로그래밍의 환경에 대해 분석해보며 그중에서도 SKT에서 채택한 플랫폼인 GVM을 기반으로한 포트리스 모바일 게임을 개발하였다. 게임은 모바일 C로 구현하였으며, 모바일 C의 특징인 Event Driven방식 게임의 실행과정을 flowchart로 나타냈다.

램이 실행될 CPU에서 동작되는 기계어로 만들어 실행시키는 것이다. 이 기술을 이용한 무선인터넷 플랫폼이 퀄컴의 브루 플랫폼이다. 표1은 모바일 플랫폼 현황을 나타낸 것이다.

1. 서 론

표1. 모바일 플랫폼 현황

최근에 소비자들은 언제, 어디서든 사용이 가능한 모바일 기기들에 큰 관심을 보이고 있다. 그 모바일 기기들은 손바닥 크기 밖에 되지 않지만, 기능이나 성능은 다른 어떤 것에도 뒤지지 않을 정도로 뛰어나다. 또한 인터넷 업계의 키워드가 e에서 m으로 바뀌고 있다. 이와 함께 무선인터넷, 혹은 M-Commerce라는 말이 인터넷 업계의 새로운 화두가 되고 있다. m은 Mobile, 즉, 이동성을 뜻하는 핵심키워드이다. 인터넷은 이미 우리 생활과, 산업의 일부분으로 자리 잡은 지금, 이제는 언제, 어디에서나 인터넷에 접속할 수 있어야 한다는 것이 현재 업계들의 추세이고, 우리가 원하는 방향이 아닐 수 없다. 지금부터 모바일 프로그래밍의 환경인 무선플랫폼의 기술을 분석해볼 것이며, 그중에서도 SKT의 표준 플랫폼인 GVM을 이용하여 모바일 게임을 구현해보았다. 본문의 2.1절에서는 무선플랫폼의 기술에 대해 기술하였으며 2.2절에서는 무선플랫폼 형식중 하나인 VM의 특징을 나타내었다. 2.3절에서는 GVM의 특징과 동작을, 2.4절에서는 GVM을 기반으로 한 모바일게임 포트리스를 개발하였다.

플랫폼	개발언어	수행방식	추진사	비고
KVM	Java	인터프리터 (VM)	LGT(Sun)	서비스
카티호크	Java	인터프리터 (VM)	LGT (아로마소프트)	서비스
SK-VM	Java	인터프리터 (VM)	SKT(XCE)	서비스
GVM	C/C++	인터프리터 (VM)	SKT (Sinjisoft)	서비스
MAP	C/C++	바이너리 (네이티브)	KTF(모빌탑)	서비스
Brew	C/C++	바이너리 (네이티브)	KTF(벨릴)	서비스
WIPI	Java, C/C++	바이너리, 컴파일러	무선인터넷표준화 포럼(이통 3사), TTA	국내표준
WITOP	Java, C/C++	인터프리터 (VM)	SKT	서비스

2. 본 론

2.1 무선플랫폼의 기술

기존의 무선인터넷 플랫폼은 VM 기술과 네이티브 바이너리 기술로 구별된다. 두 가지 기술은 상호배타적인 특성을 가지고 있으며 즉, 한쪽 기술의 장점이 다른 기술의 단점이 되고, 한쪽 기술의 단점이 다른 기술의 장점이 되는 것이다.

2.2 VM의 특징

VM(Virtual Machine)은 애플리케이션이 하드웨어에 영향을 받지 않고 실행될 수 있게 한 환경을 의미한다. VM은 이기종간의 자유로운 이식성과, 사용자의 추상화 API를 제공한다는 것이다. 그렇기 때문에 동일한 VM이 porting된 단말기에서는 동일한 실행 환경을 가지게 되고, 개발자는 단말기, OS개발 환경과는 무관하게 독립적인 개발환경에서 VM application개발이 가능하게 되는 것이다.

VM 기술은 말 그대로 가상 기계를 사용해 프로그램을 동작시키는 것이다. 작성된 프로그램을 소프트웨어 CPU에서 동작할 수 있는 중간 코드로 변환하면 가상 기계가 해당 코드를 실제 CPU용 기계어로 번역해 실행시키는 방식이다. 이러한 기술은 자바를 통해 일반화돼 있으며 자바 가상머신은 무선인터넷 플랫폼에도 이미 적용되어 있다. 국내에서는 SKT의 SK-VM, LGT의 자바 스테이션이 JVM을 탑재한 무선인터넷 플랫폼을 통해 서비스를 제공하고 있다. 또한 Moblie C를 이용한 GVM 기술이 국내에서 개발돼 SKT를 통해 서비스되고 있다.

VM의 등장하게 된 배경 중 하나는 핸드폰과 같은 모바일 device가 가지고 있는 소량의 메모리와 저속의 CPU에서 기인한 원천적인 문제점에 의한 것이다. 이는 고용량의 DB가 고속으로 처리되어야 하는 모바일 멀티미디어 서비스의 발전을 주춤하게 만들었고, 이러한 제한된 메모리 활용을 극대화하는 방법으로 다운로드와 삭제 반복하는 구조를 채택해야 했다. 이를 위해 단순 데이터를 넘어 독립적인 실행 구조를 갖는 애플리케이션을 다운로드하고 오프라인 상태에서 이를 반복 실행할 수 있도록 하는 것이 필요했고, 이렇게 해서 등장한 것이 바로 virtual machine, 즉 VM이다.

또 다른 기술인 네이티브 바이너리 기술은 일반적으로 우리가 프로그램을 작성하고 컴파일해 실행시키는 방법과 동일한 과정을 밟는다. 즉, 개발된 프로그램을 프로그

VM의 등장으로 인해 WAP 기반의 브라우징 서비스의 제한된 속도와 비싼 서비스 요금, 안정성 등에 대한 문제를 해결할 수 있었고, 각 이동통신사는 CDMA-2000 서비스를 개시한 시점인 2001년 5월부터 새로운 서비스를 위한 플랫폼으로서 VM을 채택하기 시작했다.

2.3 GVM(General Virtual Machine)

GVM은 모바일 C를 기반으로 하는 무선 애플리케이션 다운로드 플랫폼으로 SKT에서 채택한 가상머신을 지칭하는 것으로 단말기 소프트웨어의 UI 태스크에 위치하기 때문에 통화와 같은 단말기의 기본기능에는 영향을 주지 않는다. GVM의 특징으로는 단말기의 리소스 요구를 최소화 한 것과, 단말기에 porting이 쉽고, 모바일 C언어를 사용해 애플리케이션 개발이 용이함은 물론 실행 성능이 우수하다는 것을 말할 수 있다.

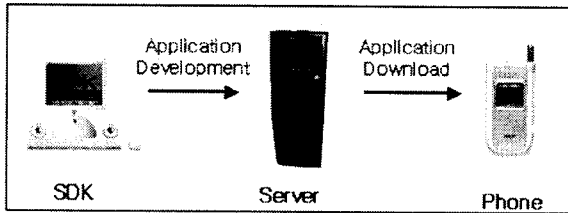


그림 1. GVM시스템의 구성

그림 1에서 확인할 수 있듯이 GVM은 SDK, Server, GVM모듈이 탑재된 핸드폰으로 구성된다. GVM SDK(Software Development Kit)는 GVM이 탑재된 단말기에서 실행되는 애플리케이션을 작성하는 개발도구를 말하며, 모바일 C컴파일러, 미디어툴킷, 에뮬레이터로 구성된다. Server는 애플리케이션 DB를 보유하고 단말기와 접속하여 애플리케이션의 다운로드를 제공하는 기능을 수행한다. GVM모듈은 단말기에 탑재되는 S/W를 말하며, Porting의 용이성이 고려되어 제공이 되며, Event-Driven방식으로 실행하도록 설계된다. 게임 개발자는 SDK를 이용하여 모바일 게임을 제작하고 Server에 등록을 하고 사용자는 Server에 접속해서 Download 받아 사용하게 된다.

2.3.1 GVM SDK의 구성요소

SDK는 크게 모바일 C컴파일러, 미디어툴킷, 에뮬레이터로 구성된다.

모바일 C컴파일러는 모바일 C로 작성된 프로그램을 컴파일하여 오브젝트파일(*.SGS)을 생성한다. SGS파일은 단말기에 다운로드 되어 가상머신에 의해 실행된다. 또한 이것은 에뮬레이터상의 실행파일이 되어 에뮬레이션 할 수 있게 된다. SGS 파일은 내부적으로 헤더영역 코드영역, 데이터 영역으로 구성된다.

컴파일과 빌드를 수행하였을때 compile error나 warning이 없으면 File Conversion Success!!! 라는 메시지가 뜨며, 그 소스파일에 사용된 String Data, Image Data등의 개수를 확인할 수 있다. 여기서 사용된 Image Data는 7개임을 알 수 있다.

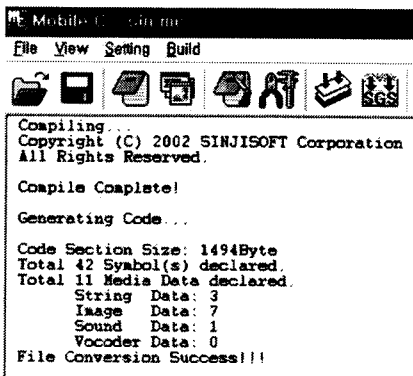


그림 2. Mobile C Compiler 실행화면

미디어 툴킷은 다시 오디오 툴과 이미지 툴로 나눌 수

있다. GVM SDK에서 제공하는 오디오 툴은 Buzzer Master와 Audio Master가 있다. Buzzer Master를 이용해 음원을 직접 제작하여 모바일 C 프로그램에서 사용할 수 있는 형태로 음원 데이터를 변화 시킬 수 있다. 또한 Audio Master는 이미 작성된 음원 데이터를 읽어 모바일 C형태로 변환시켜 준다. 16화음인 MA2음원과 40화음인 MA3음원을 Audio Master를 사용해 converting하면 모바일 C 형태인 *.ssd 파일로 변환되는 것이다. 이렇게 변환된 음원은 소스 프로그램에 include하여 사용하며, 게임 속의 음원은 이러한 과정을 통해서 얻게 되는 것이다.

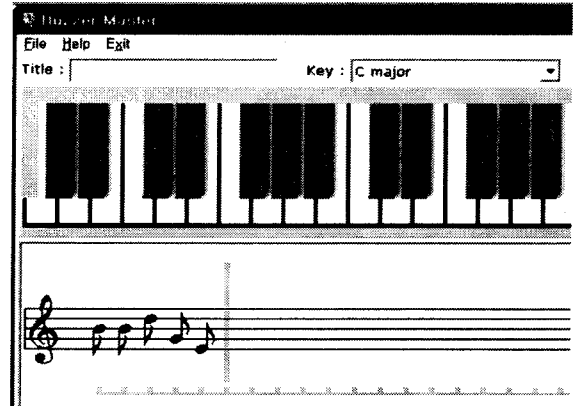


그림 3. Buzzer Master 실행화면

다음으로 GVM의 이미지 제작과정을 살펴보겠다. 이미지 제작에 사용되는 툴로는 Image Master가 있다. Image Master의 사용방법은 먼저 포토샵, 페인트샵 등의 그래픽 툴을 이용하여 *.bmp, *.gif, *.png 등의 원본파일을 제작한다. 그리고 Image Master에서 원본 이미지 파일을 읽어 들여 모바일 C형태인 *.sbm파일로 변환하게 되는 것이다. 이것 역시 모바일 C소스 프로그램에 include되어 사용된다.

그림 5는 하나의 그림파일을 불러온 것이다. 하나의 그림파일 속에 3개의 탭 그림이 있는데 그 3개의 그림 중 사용하고자 하는 그림에 마우스로 드래그 하여 선택을 하고 그것의 이름을 적어준다. 이름을 적지 않았을 경우 배열 형식으로 저장이 되며 소스프로그램에서 이미지를 사용할 때는 이름으로 불러서 사용하게 된다. 즉, 하나의 파일을 include했지만 3개의 그림을 각각의 이름으로 사용하는 것이다.

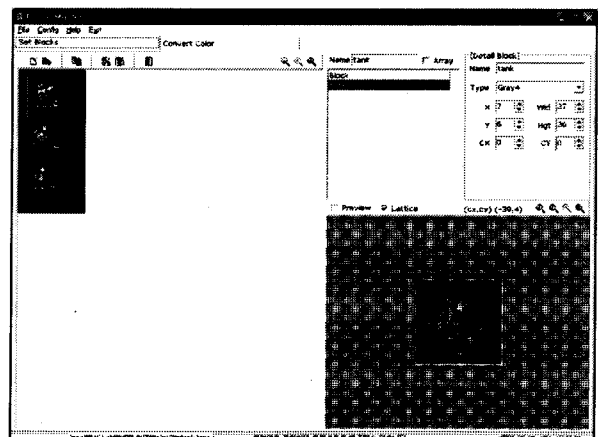


그림 4. Image Master 실행화면

GVM 에뮬레이터는 PC에서 GVM 애플리케이션의 동작을 단말기와 동일한 상황으로 시뮬레이션하고, 단말기의 다양한 하드웨어 사용, RTOS의 이벤트 레벨까지 에뮬레이션하여 단말기와 동일한 조건으로 동작 시험을 가

능하게 한다.

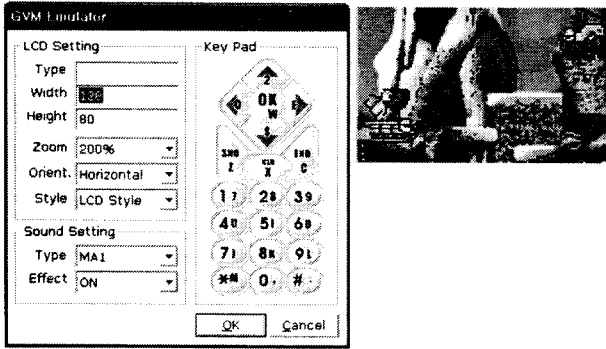


그림 5. 에뮬레이터 실행화면

2.3.2 GVM SDK의 동작

앞서 살펴본 것과 같이 Sound와 Image는 각각 compile되어 모바일 C에서 사용가능한 형태로 변환되며 이것은 프로그램 소스파일에 include하여 사용한다. 소스프로그램은 Mobile C Compiler를 통하여 GVM의 오브젝트 파일인 sgs로 컴파일되며, 이 파일로 에뮬레이터의 시뮬레이션이 가능해지며, GVM이 탑재된 단말기에서 다운로드 받아서 사용하게 되는 것이다. 이것의 흐름을 그림6에 나타내었다.

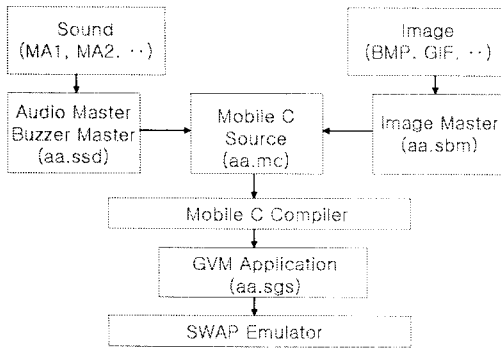


그림 6. GVM SDK의 동작

2.3.3 GVM이 제공하는 기능

GVM은 서버에 접속하여 원하는 애플리케이션을 다운로드해 실행할 수 있게 한다. 일단 다운로드한 애플리케이션은 단말기의 메모리에 저장되어 사용자가 삭제하기 전까지는 재 다운로드 없이 수행할 수 있다. 이것은 VM의 등장하게 된 요소이며, 가장 큰 특징이라 할 수 있다. GVM은 모든 종류의 단말기 LCD 및 오디오 장치를 제어할 수 있으며, 다양한 이미지 포맷을 수용한다. 또한 GVM은 타이머를 이용한 리얼타임 애플리케이션을 실행할 수 있고, 단말기에서 TCP/IP소켓 서비스를 제공하는 경우 이를 이용하여 클라이언트-서버 형식의 애플리케이션을 실행할 수 있다. 그리고 GVM에서는 단말기의 Vibrator, KeyTone, BackLight등의 장치를 제어할 수 있는 API를 제공하므로 이를 이용해 생동감 있는 애플리케이션을 실행할 수 있다. 마지막으로 GVM이 제공하는 기능으로는 시리얼 통신을 가능하게 한다는 것이다. 단말기의 RS-232C 포트를 통해 컴퓨터와 같은 외부 장치와 연결하여 데이터를 주고받을 수 있는 응용 기능이 제공된다.

2.3.4 Mobile C의 특징

첫째로 표준 C가 철차적 언어임에 반해 모바일C는 이벤트 드리븐 방식으로 실행된다. 어플리케이션이 시작,

키패드 입력, 타임아웃발생등과 같은 이벤트 핸들러의 집합으로 구성된다. 또한 함수의 재귀순환이 금지되어 있고, data type을 2byte integer로 제한한다. 표준 C에는 없는 image, sound, voc, string과 같은 미디어 type이 추가되었다.

포트리스 게임에 사용된 이벤트 핸들러로는 EVENT_TIMEOUT, EVENT_KEYPRESS가 있다. EVENT_TIMEOUT은 setTimer에서 지정한 일정시간이 경과했을 때마다 Timeout Event를 발생시켜 해당되는 이벤트를 수행하는 것이다. EVENT_KEYPRESS는 사용자가 단말기에 키를 누를 때 발생하는 이벤트이다. 이벤트 발생시 사용자가 누른 키 정보는 swData 값으로 알 수 있는데 1 입력 되었을 때의 swData는 SWAP_KEY_1 또는 0x01이다. 이 값은 SString.h에 정의 되어 있다. SString.h은 GVM SDK에서 제공되는 헤더파일로, LCD를 제어하는 함수, 진동을 제어하는 함수 등 여러 함수들에 관해 정의 되어 있으며, 소스파일을 제작하기 전에 항상 include하여야 한다. swData 값을 이용해서 switch-case 문으로 각각의 키를 나누어서 사용자가 누른 키에 해당되는 이벤트를 수행하게 되는 것이다.

2.4 모바일 게임의 구현

모바일 C를 이용해 포트리스 게임을 구현해 보았다. 이 게임은 왼쪽 탱크 캐릭터에서 미사일이 발사되고, 날아가는 미사일의 좌표와 오른쪽 탱크 캐릭터의 좌표가 같아지면 탱크가 터지고 CLEAR!! 라고 출력함과 동시에 count하여 stage수를 출력하게 구현되었다.

그림 7은 처음 게임을 실행시켰을 때의 에뮬레이터상의 모습이다. 그림 8은 미사일을 쏘았을 때이고, 그림 9는 미사일이 오른쪽 탱크에 맞았을 때 탱크가 터지는 모습을 capture 한 것이다. CLEAR!! stage: 1이 출력되었음을 확인할 수 있다



그림 7. 초기화면



그림 8. 포탄발사



CLEAR!! stage: 1
그림 9. 명중

2.4.1 포트리스 게임의 순서도

포트리스 게임을 크게 3등분하면, main()함수를 비롯한 게임을 구성하고 동작시키는 여러 함수들과, 키패드 입력 시 발생하는 이벤트 처리 부분과, 타임아웃이 발생하였을 때 처리해주는 부분으로 나눌 수 있다. 구현 게임의 전반적인 동작 알고리즘을 그림 10에 도시하였다. 개발된 프로그램은 GVM은 서버에 등록하기 전에 SDK에서 제공하는 emulator로 실제 핸드폰에서 동작하는 것과 같은 환경을 만들 수 있다. emulator를 실행시키면 main()함수안의 initFunc()가 호출되어 모든 변수들을 초기화시켜주고 게임이 실행되었을 초기모습을 그려주게 된다.

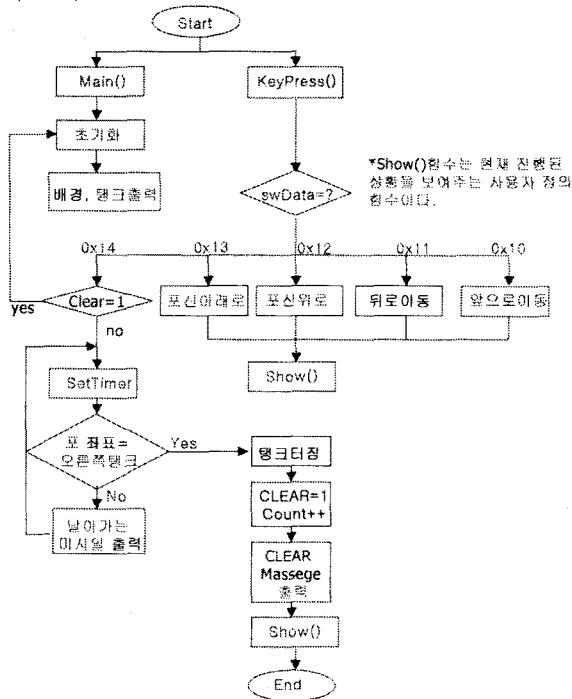


그림 10. 포트리스 게임의 동작 알고리즘

여기서 KeyPress(swData)는 게임을 조작할 때 사용하는 핸드폰상의 키패드 입력 시 발생하는 이벤트로, swData는 프로그래밍 할 때 include시키는 SScript.h 헤더파일 안에 1부터 #까지의 키가 0x10 또는 SWAP_KEY_LEFT의 형태로 정의되어있다. swData = 0x14일 때는 OK key를 눌렀을 때를 나타낸다. 이때 미사일이 발사되면서 SetTimer가 동작되는데, 이것은 지정해준 시간이 지나면 Timeout을 발생시키게 되고 미사일이 발사하게 된다. 미사일 각도를 조절하여 오른쪽 캐릭터에 발사하여 맞혔을 경우 count를 하여 stage를 출력하고, 게임을 다시 시작할 경우 OK key를 누르면 새로 초기화하고 다시 게임이 시작되게 되는 구조로 코딩하였다.

3. 결 론

신지소프트에서는 GVM의 워 버전인 GNEX를 출시했다. GNEX는 512kB로 GVM보다 사용가능한 메모리가 클 뿐만 아니라 3D까지 지원되기 때문에 앞으로 좀더 눈이 즐거운 게임을 즐길 수 있게 될 것이다.

VM과 SKT에서 채택한 플랫폼인 GVM에 대해 전반적으로 알아보고 모바일 C를 이용해 간단한 프로그램을 구현해 보았다. 표준 C와는 다른, 모바일 C의 특징인 이벤트처리 부분인 KeyPress()나, Timeout()등을 코딩해 보면서 모바일 C와 표준 C의 차이점 그리고 flowchart로 게임이 동작되는 과정을 살펴보았다. 향후에는 게임뿐 아니라 영어학습 프로그램, 원격제어프로그램 등에

관한 연구를 수행할 예정이다.

[참 고 문 헌]

- [1] 김정훈, "Mobile Game Contents Project", 베스트북, 2002.
- [2] 이정환, 수주호, "GVM&MAP 모바일 프로그래밍", 대림, 2002.
- [3] 앤슬래시닷컴 저, "GVM Programing", 삼양출판사, 2001.
- [4] 김상택, "Mobile Contents 발전방안", Telecommunications Review, 제10권 6호, pp.1124-1131, 2000.
- [5] 윤소란, "모바일 게임 시장 및 개발 동향", 한국정보처리학회, 제9권 3호, pp. 42-48, 2002. 5.
- [6] 김승훈, "GVM 개발환경에서 모바일 온라인컨텐츠를 위한 프로토콜", 멀티미디어학회논문지, 제 7권 2호, pp. 241-250, 2004. 2.
- [7] 황병연, 오명석, "모바일 게임을 위한 압축기술", 정보처리학회지, 제9권 3호, pp. 49-54, 2002. 5.