

GVM 과 MobileC 를 이용한 모바일 게임 프로그래밍

김필진*, 남동근, 오세만
동국대학교 컴퓨터공학과
e-mail : {kpj0817,capsrom, smoh}@dongguk.edu

Mobile Game Programming Using GVM and MobileC

Pil-Jin Kim*, Dong-Geun Nam, Se-Man Oh
Dept. of Computer Engineering, Dongguk University

요 약

컴퓨터 게임의 발달은 오프라인 PC 게임에서 인터넷의 보급으로 인해 온라인 게임으로 발전하였다. 그 후 인터넷 환경은 유선 환경뿐만 아니라 무선 환경에서도 무선 단말기를 이용하여 게임 서비스가 가능한 수준에 이르렀다. 그러나 적은 메모리와 낮은 수준의 처리능력을 가지고 있는 제한적인 모바일 환경의 특성으로 유선 인터넷에서와 같이 멀티미디어를 포함한 다양한 형태의 정보제공 및 인터랙티브한 모바일 게임이 어려운 현실이다.

본 논문에서는 WAP(Wireless Application Protocol)을 기반으로 하여 다운로드 솔루션을 채택한 GVM(General Virtual Machine) 상에서 실행될 수 있는 모바일 게임을 설계하고 MobileC 언어를 사용하여 제작하였다. 또한, 모바일 게임을 설계하고 제작하는 과정을 통해서 단말기의 제한적인 실행환경에서 최대한의 효과를 나타낼 수 있는 개발 모델을 제시하고자 한다. GVM 은 무선 단말기 상에서 동적인 응용프로그램을 실행할 수 있는 가상기계 플랫폼이다.

1. 서론

최근 몇 년간 인터넷을 위한 여러 분야의 기반시설이 급속히 발전해 오고 그에 따라 게임의 구도도 인공지능을 갖춘 PC 게임에서 온라인 네트워크라는 기술적 개념이 포함된 온라인 게임으로 변화되었다. 이는 단순 인공지능 게임 프로그램간의 대결이 아닌 인간간의 실시간 대결이 가능했다는 점에서 많은 사용자들의 흥미를 유발 시키게 되었고[1] 그 후 1999년 중반부터 WAP의 등장으로 무선 인터넷 서비스가 가능해지면서 제한된 공간에서 벗어나 장소에 구애를 받지 않고 게임을 즐길 수 있게 되었다. 현재에는 가상기계를 무선단말기에 탑재하여 게임을 비롯한 응용 프로그램을 동적으로 다운로드하여 실행시켜주는 가상기계 솔루션이 출현하였다. 이 가상기계를 이용한

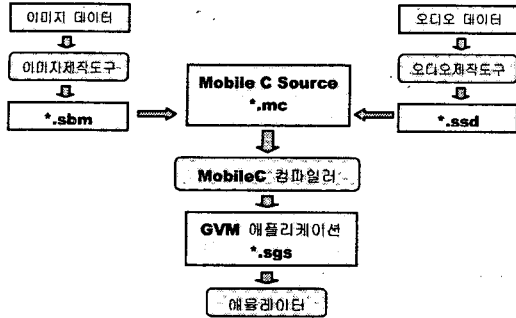
무선 서비스는 모든 플랫폼에 독립적으로 실행된다는 장점을 갖고 있다. 이와 같은 장점으로 현재 국내 모바일 게임 시장에서 GVM[4]과 KVM[6]등의 가상기계를 이용한 게임 콘텐츠를 제공하고 있으나 아직까지는 망 속도의 제한, 잦은 접속 중단, 단말기의 제한 등의 단점을 갖고 있다. 본 논문은 이러한 문제점을 극복하는 방법을 모색하기 위해 가상기계 솔루션중 GVM 과 MobileC 를 이용하여 무선 인터넷 게임을 설계하고 구현하였다.

2. 관련연구

2.1 GVM SDK

GVM SDK(GVM Software Development Kit)는 GVM 애플리케이션을 작성하는 프로그램 개발도구를 총

칭하며 다음과 같은 구성도를 갖는다.



[그림 1] GVM SDK의 구성

[그림 1]은 GIF, BMP 이미지를 VDI(Variable Depth Image) 형식으로 정의하고 MobileC 소스 형태로 변환하는 이미지 변환기, Buzzer, MA1, MA2, EVRC 등의 오디오 데이터를 MobileC 소스 형태로 바꾸주는 사운드 편집기, 그리고 MobileC 컴파일러 및 PC에서 작동하는 GVM 에뮬레이터로 구성 되어있다.

(1) 다운로드 시스템

다운로드 시스템은 GVM SDK, Application Download와 Network Client Application의 연결을 수행하는 서버와 GVM Module이 내장된 단말기로 구성되어 있다.

GVM Module은 단말기에 탑재되는 가상기계와 관련된 Library Module을 뜻하며 SVM(SWAP Virtual Machine), MIL(Mobile Interface Library), MIDD(Mobile Interface Device Driver)로 구성된다.

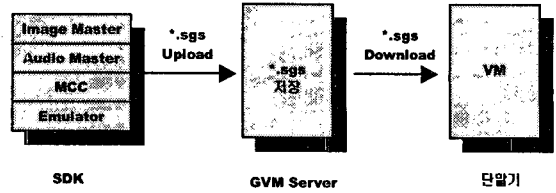
응용프로그램 개발시에 단말기의 속성은 LCD 크기, 콘텐츠의 컬러, 그리고 콘텐츠의 크기 등을 결정하는 기준이 된다.

(2) MobileC

MobileC[4]는 표준 C언어를 기반으로 제한된 모바일 플랫폼에 맞추어 응용프로그램을 작성할 수 있는 언어이다. MobileC 프로그램은 MobileC 컴파일러를 통해 가상기계인 GVM에서 실행될 수 있는 코드(*.sgs)로 변환된다.

2.2 실행방법

제작된 이미지와 사운드를 GVM SDK의 이미지와 사운드 제작도구를 통해 GVM 규격으로 변환하여 MobileC 소스파일(*.mc)에 포함시킨 후 컴파일하여 *.sgs 파일을 생성한다. 이렇게 작성된 모바일 응용프로그램은 서버에 업로드 시킨 후 GVM이 내장된 단말기로 다운로드되어 실행되게 된다. [그림. 2]는 이와 같은 과정을 잘 나타내고 있다.



[그림 2] GVM 게임 실행 모델

3. 게임의 설계

3.1 게임의 기획

모바일 게임의 기획 단계는 그 성격상 단말기의 성능에 큰 비중을 두고 기획하게 된다. 단말기의 LCD 크기, 메모리의 용량 등이 실제 단말기에서의 게임 실행에 큰 영향을 미치는 중요한 요소가 되기 때문이다.

3.1.1 스토리보드

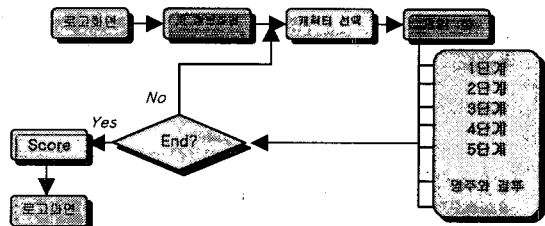
일반 PC 온라인 게임과 달리 모바일 게임의 스토리 보드는 적당한 수의 캐릭터, 단순하면서도 누구에게나 흥미를 유발시킬 수 있는 내용의 시나리오, 그리고 간단한 조작법등을 고려해서 작성해야 한다. 본 논문에서 제작된 게임의 스토리보드를 간단히 정리하였다.

구분	장르
장르	슈팅
제목	마법사의 돌
등장인물	짜르(남주인공), 미미(여주인공), 마법사
시나리오	옛날 시골마을의 영주가 된 마법사는 마을 사람들에게 막대한 세금을 요구한다. 이에 대항하는 짜르와 미미. 마법사는 마법을 부려 하늘에서 각종 물체를 떨어뜨려 농사를 망치려 한다. 주인공들은 그들의 무기로 떨어지는 물체를 파괴하여 농지를 보호한다.

[표 1] 스토리보드

3.1.2 게임의 구성

본 논문에서 제작한 「마법사의 돌」 게임의 구성은 아래와 같다.



[그림 3] 「마법사의 돌」 게임의 구성도

일반 PC 게임의 경우 게임이 끝난 후 그 정보를 서버에 저장하거나 개인 PC에 파일로 저장하여 보관한 후 그 부분을 다시 연결하여 게임을 계속할 수 있다.

하지만 모바일 게임의 경우 이 요소는 게임을 로딩하는데 있어서 상당한 지연을 초래하므로 대부분 제외되고 있다.

3.2 게임 사양

GVM 과 MobileC 를 이용하여 게임을 개발할 때 미리 결정해야 하는 몇가지 중요한 사양들이 있다. 다음 [표 2]는 「마법사의 돌」에 대한 게임 사양이다.

구분	내용
LCD Size	128 * 106
Image	256 Color
Sound	Buzzer Sound
Key 할당	1:위로, 7:아래로, 5:슈팅 취소:메인메뉴, 확인: OK

[표 2] 「마법사의 돌」 게임 사양

3.3 이미지와 사운드 디자인

이미지의 디자인은 LCD 크기를 고려하여 사용자가 그 형태를 인지할 수 있는 최소의 크기로 디자인하는 것이 좋다. 여기서는 LCD 크기가 128*106 임을 고려하여 각각의 이미지 크기가 20*20 을 넘지 않도록 디자인하였다. 사운드는 Buzzer 와 MA1, MA2 등을 사용하는데 GVM SDK 중 오디오 제작도구인 Buzzer Master 를 이용하여 Buzzer 사운드를 디자인했다.

4. 게임의 구현

4.1 이미지와 사운드 제작



[그림 4] 「마법사의 돌」 게임에 사용된 이미지들

이미지는 ImageMaster 상에서 블록 단위로 사용하게 되는데 한 블록 안에 몇 개의 컬러를 사용했는지에 따라서 용량에 큰 영향을 미치게 된다. 가령 20*20 크기의 오브젝트에 16 개의 컬러를 사용할 경우 이미지 블록은 color4 로 셋팅되어 20*20*4(bit) = 1600 비트의 용량을 차지한다. 하지만 17 개의 컬러를 사용한다면 color8 로 셋팅되어 20*20*8(bit) = 3200 비트의 용량을 차지하게 된다. 따라서 각 이미지 오브젝트마다 적절한 컬러수의 선택이 매우 중요하다. 제작된 이미지는 이미지 제작도구를 이용해 소스코드 형태로 변환하여 사용한다.

사운드는 BuzzerMaster 를 이용해 *.pcm 파일을 제작한 뒤 오디오 제작도구를 이용해 소스코드 형태로 변환하여 사용한다.

4.2 프로그래밍

MobileC 언어에는 Header Section, Event Handler 등의 특수한 기능이 있다. [그림 5]와 같이 Header Section 에서는 게임의 사양을 정의한다.

```

%{
#define SCRIPTVER      1          // script ver
#define LCDCCLASS      255       // All size
#define IMAGETYPE      4         // 256 Color
#define AUDIOTYPE      1         // Buzzer
#define SCRIPTTYPE     1         // Stand-Alone
...
// 0000101FF040101.sgs(Buzzer)
// CPID/ScriptVer/LCDCclass/Img/Snd/ScriptType
}%
    
```

[그림 5] 헤더 정보

스크립트 파일의 파일명은 Header Section 정보의 조합으로 구성함으로써 서비스에 다른 게임 콘텐츠들과 구별될 수 있도록 한다.

기존의 C 프로그램과 비교하여 MobileC 프로그램의 가장 큰 특징은 Event-Driven 구조라는 점이다. MobileC 에는 EVENT_START(), EVENT_TIMEOUT(), EVENT_KEYPRESS() 등의 Event Handler 가 있다.

```

void main()
{
    ClearWhite();           // LCD 버퍼초기화
    Init();                 // 초기화
    baseX = (swWidth - 128)/2; // 상대좌표기준값
    baseY = (swHeight - 106)/2;

    gm_state = STATE_LOGO; // 게임상태
    PlaySound(sd_logo);    // 사운드
    SetTimer(INTERVAL, 1); // 반복타이머
}
    
```

[그림 6] EVENT_START()

EVENT_START()는 C 프로그램의 main 함수와 똑같은 기능을 수행하는 부분으로서 게임 수행시 처음으로 실행되는 부분이다. [그림 6]에서 baseX 와 baseY 는 이미지를 상대좌표로 그려주기 위한 변수이다. 모바일 게임은 다양한 LCD 크기에 대응해야 하므로 게임 제작시 기존 LCD 크기를 정한 후 그와 다른 LCD 크기에서도 같은 비율로 게임이 디스플레이 될 수 있도록 상대좌표를 사용해서 프로그래밍한다.

EVENT_KEYPRESS()는 키입력이 발생했을 때 수행되는 Event Handler 이다. 키정보는 시스템 변수인 swData 에 저장이 된다. 게임의 상태를 지정하여 각 게임 상태별로 swData 에 따른 실행코드를 프로그래밍한다.

EVENT_TIMEOUT()은 SetTimer()에서 설정한 타이머가 호출될 때마다 수행되는 Event Handler 이다. 즉 각 프레임마다 수행되는 부분으로서 게임진행에 가장 중요한 부분이다. 정의된 함수들을 수행하고 Back 버퍼에 이미지들을 그려준 뒤 마지막으로 Flush()함수

를 이용하여 Back 버퍼의 이미지를 Front 버퍼에 그려준다.

PC 에 비해 제한적인 실행환경을 가진 무선 단말기에서 게임 콘텐츠가 정상적으로 실행될 수 있도록 하기 위해서는 용량과 실행속도를 최적화 시키는 프로그래밍이 매우 중요하다. 다음 [그림 7]은 이미지 용량을 줄이기 위한 코드이다.

```

Void DrawGame()
{
    ClearWhite();

    CopyImage(baseX, baseY, img_back);
    CopyImageDir(baseX, baseY, img_back, S_DIR_MIRROR);
}
    
```

[그림 7] 대칭이미지 이용 코드

CopyImage 는 이미지 블록을 LCD 버퍼에 그려주는 일을 하고 CopyImageDir 은 해당 이미지를 지정된 방향으로 버퍼에 그려주는 일을 수행한다. 즉, 원이나 사각형 등의 좌우가 대칭되는 모양을 가진 이미지는 절반만 사용해도 전체이미지를 모두 사용하는 것과 같은 효과를 낼 수 있다. [그림 7]에서는 배경이미지를 좌우가 대칭되도록 제작하여 LCD 버퍼에 이를 대칭하여 그려줌으로써 배경 이미지의 용량을 절반으로 줄였다.

다음 [그림 8]은 LCD 버퍼를 이용해서 실행속도를 줄이는 코드이다.

```

Void DrawGame()
{
    ClearWhite();
    if(first_draw){
        CopyImage(baseX, baseY, img_back);
        SaveLCD();
        first_draw = FALSE;
    }
    else
        RestoreLCD();
}
    
```

[그림 8] LCD 버퍼 이용 코드

프레임마다 반복되어 나타나는 이미지가 있는 경우에 이것을 LCD 버퍼에 그려준다는 것은 실행시간의 낭비를 초래한다. 우선 반복되는 이미지를 SaveLCD() 함수를 사용해 Back 버퍼에 저장해 둔 후 다음 프레임에서 Back 버퍼에 저장한 이미지를 RestoreLCD() 함수를 사용해서 현재의 LCD 버퍼로 가져온다. 게임 실행시 대부분의 시간이 이미지를 그리는데 할당된다는 점에서 Back 버퍼의 사용은 실행속도와 밀접한 관계가 있다. [그림 8]의 코드에서는 매번 반복되는 배경 이미지를 최초의 한번만 그려준 뒤 Back 버퍼에 저장한 후 반복적으로 사용하고 있다.

4.3 실행파일 생성

소스파일(*.mc)에 이미지 변환파일(*.sbm)과 사운드 변환파일(*.ssd)을 포함한 후 MobileC 컴파일러로 컴파일하면 실행파일(*.sgs)이 생성된다. 실행파일은 에뮬레이터와 실제 단말기에서 실행되는 파일이다. 실행파일의 용량은 그레이(gray) 게임 콘텐츠인 경우에는 48k, 컬러 게임 콘텐츠인 경우에는 128k 를 넘지 못하도록 되어있으므로 항상 실행파일의 크기를 정확히 체크해야한다.

4.4 테스트

제작된 게임은 에뮬레이터나 실제 단말기를 통해서 테스트할 수 있다. 에뮬레이터는 PC 상에서 단말기와 동일한 환경에서 실행파일을 수행할 수 있도록 해준다. 그러나 PC 와 단말기는 하드웨어의 차이가 크기 때문에 실행 속도면에서 많은 차이를 보일 수 있다. 따라서 반드시 단말기 테스트를 거쳐야 한다. 컬러가 다양하고 게임의 복잡도가 높을수록 PC 와 단말기에서의 차이는 더욱 커진다.

5. 결론 및 향후연구

본 논문에서는 GVM 플랫폼에서 실행될 수 있는 「마법사의 돌」이라는 게임 콘텐츠를 설계하고 MobileC 언어를 이용하여 제작하였다. 무선단말기의 다양한 LCD 크기와 적은 메모리, 그리고 전송속도 등이 구현의 제한요소가 되었다. 이와 같은 한계점을 극복하기 위하여 본 논문에서는 이미지 용량의 최적화, 상대좌표를 이용한 프로그래밍, 그리고 Back 버퍼를 사용한 실행시간의 단축 등의 기법을 적용하였다.

앞으로는 휴대용 단말기의 발전과 차세대 통신 서비스 기술인 IMT-2000 의 보급으로 인해 무선단말기의 제한요소가 개선되고 폭 넓은 콘텐츠 제작을 할 수 있을 것이다. 모바일 게임 콘텐츠도 PC 게임의 수준을 요구하게 될 것이며 이에 따른 개발 모델의 지속적인 연구가 필요하다.

참고문헌

- [1] 이만재, “온라인 게임 엔진 기술 동향”, 정보과학회지, 제 20 권 제 1 호, pp.12-18, 2002.
- [2] 김현준, “무선인터넷상에서 멀티미디어 서비스와 가상현실의 설계 및 구현”, 정보처리논문지, 제 8 권 제 2 호, pp.163-166, 2001.
- [3] 장인결, “PDA 용 무선 게임엔진과 개발도구의 설계”, 정보처리논문지, 제 8 권 제 2 호, pp.225-228, 2001.
- [4] GVM 기술지원 홈페이지, <http://www.gvmclub.com>
- [5] 모바일 프로그래밍 관련 포털 사이트, <http://mobilelab.co.kr>
- [6] KVM 관련 정보 지원 개인 홈페이지, http://my.netian.com/~bless2k/jbible2_supp_KVMnGVM.htm/
- [7] 모바일 관련 업체 기술 소개 및 장비소개 사이트, <http://www.intromobile.com/>
- [8] 앤슬래시닷컴, GVM Programming, 삼양출판사, 2001.