

안드로이드 NDK(Native Development Kit)를 이용한 3D 큐브 게임 이식 및 구현

고은별*, 김녹희*, 황성미*, 이종우**

요약

휴대폰 사용자들 대부분이 스마트폰으로 옮겨가면서 사용자들은 이제 스마트폰으로 다양한 게임을 즐기고 있다. 그런데 현재 앱스토어에 기기의 성능을 고려한 2D 게임은 다양하게 올라오고 있지만 3D 게임의 수는 아직도 많이 부족한 것이 실정이다. 이는 3D 게임이 많은 컴퓨팅 성능을 필요로 하기 때문인데, 본 논문에서는 안드로이드에서 NDK를 이용하여 3D큐브 게임을 구현하였다. 기존 비주얼C++ 일리히트 3D 큐브 소스를 NDK를 이용해 안드로이드 시스템 라이브러리로 이식하고 편의 기능을 추가하는 방식으로 구현하였다. NDK를 이용했으므로 기존 C++ 코드의 대부분이 자바 코드로 변환되지 않고 그대로 실행될 수 있어서 실행 시험 결과 비교적 구형 스마트폰 상에서도 3D 그래픽 동작들이 무난하게 이루어짐을 확인할 수 있었다.

키워드 : 안드로이드, NDK, 3차원 큐브 게임, 이식

Porting and Implementation of a 3D Cube Game using Android NDK(Native Development Kit)

Eunbyul Koh*, Nokhee Kim*, Sungmi Hwang*, Jongwoo Lee**

Abstract

Almost all the mobile phone users already moved or are now moving away to smartphones for their various applications like games. If we are to speak about game applications, due to the performance limits of smartphones, 2D games are predominant over 3D games in every app. store. In this paper, we implement a 3D cube game application by porting an existing visual c++ irrlicht cube application to android platform library using the android Native Development Kit. After the porting is done, we add a few new features for more fun. Because the android NDK makes the existing C/C++ codes run directly on the android operating systems, we found by real execution tests that our 3D cube app. is well executed on a low-end android smartphone without any performance problem.

Keywords : Android, NDK, 3D Cube, irrlicht, Porting

1. 서론

스마트폰이 출시되면서 사람들의 생활중심은

하루가 다르게 스마트폰으로 옮겨 가고 있으며 심지어 어린이 장난감까지도 모바일 기기로 나오는 추세이다. 또한 하루가 멀다 하고 출시되는 핸드폰기와 업데이트되는 운영체제 버전으로 인해 사용자들의 그래픽 응용에 대한 요구도 날로 높아지고 있다.

하지만 스마트폰에서 지원할 수 있는 그래픽 기능은 아직까지 제한되어 있고, 또한 고품질의 그래픽 화면을 구현하기에는 여전히 많은 한계가 있다[1]. 안드로이드 앱은 보통 객체지향 언어인 자바(JAVA)로 개발되는데 자바는 웹 활용

※ 교신저자(Corresponding Author): Jongwoo Lee
접수일:2013년 09월 24일, 수정일:2013년 09월 30일
완료일:2013년 09월 30일
* 숙명여자대학교 멀티미디어학과
** 숙명여자대학교 멀티미디어학과 교수
Tel: +82-2-710-9952, Fax: +82-2-710-9704
email: bigrain@sm.ac.kr

능력은 뛰어나지만 게임을 구현하기에는 처리 성능 측면에서 많은 한계를 가지고 있다.

스마트폰용 OpenGL/ES는 PC용 3D 그래픽스 엔진인 OpenGL의 서브셋으로 스마트폰이나 PDA 등과 같은 임베디드 기기를 위한 API 이다. 따라서 OpenGL에 비해 지원하는 API 타입의 수가 적고 즉시 모드를 지원하지 않으며 텍스처 맵핑은 2D만 지원하는 등의 단점이 있다.

그래서 본 논문에서는 안드로이드 NDK(Native Development Kit)를 이용하여 목표하는 게임을 구현하였다. NDK를 이용하면 JNI(Java Native Interface)를 통해 안드로이드 커널 위의 C 라이브러리를 이용할 수 있어 성능이 있기 때문이다. 또한 OpenGL/ES의 단점을 보완하면서도 상대적으로 가벼운 안드로이드용 그래픽 라이브러리인 irrlicht[2]를 사용하였다. Irrlicht는 C++ 다중 플랫폼 3D 그래픽 라이브러리로써 게임이나 과학적 시각, 2D/3D 앱을 위한 API를 지원한다.

본 논문에서는 안드로이드 NDK와 irrlicht 라이브러리를 이용해 '똑똑이 큐브'라고 이름 지은 3D 큐브 게임을 구현하고 그 구현 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 현재 관련기술 동향을, 3장에서는 구현 환경과 구현 과정을, 4장에서는 위에서 설명한 기술들을 이용하여 구현한 똑똑이 큐브 게임의 실행화면을 기술한다. 마지막으로 5장에서는 결론을 맺는다.

2. 관련 기술 및 관련 연구

2.1 안드로이드 NDK 동향

안드로이드 NDK는 안드로이드 애플리케이션(이하 앱)에서 C 와 C++ 같은 네이티브 코드를 사용할 수 있게 하는 도구이다. 즉 JNI를 통해서, SDK가 기본적으로 지원하지 않은 기능들을 추가 개발해서 상위 자바 응용 단에서 쓸 수 있게 해줌으로써 게임과 같이 속도에 민감한 앱을 개발하는 데에 도움을 주는 도구인 셈이다.

하지만 현재 NDK를 활용한 앱 개발은 그리 활발한 편은 아닌데, 그 이유는 비교적 배우기 쉬운 자바 외에 C/C++ 언어를 알아야 한다는 점과 가상머신 상에서 수행되는 앱의 성능 저하를 극복하고자 하는 필요성이 있어야 하기 때문인 것으로 판단된다. 또한, NDK를 사용하여 고품질 그래픽 요소를 지원하는 모바일 게임 앱을 제작하려 해도 모바일 기기의 성능 자체가 따라주지 못한다면 상용화하는데 큰 걸림돌로 작용하기 때문일 것이다.

하지만 하루가 다르게 발전하는 모바일 기기의 성능은 최근 들어 노트북 못지않은 성능과 메모리 크기를 가지게 되었다. 또한 하루에도 800만개의 앱이 쏟아지는 요즘, 사용자들의 고품질 그래픽 앱에 대한 요구도 날로 증가 하고 있다 하겠다.

본 연구를 진행했던 2012년 3월 ~ 7월 말 동

<표 1> NDK 버전별 차이점 요약

ndk r3	ndk r4	ndk r7
ndk폴더/apps/폴더/프로젝트 배치	프로젝트 배치 상관없음	
폴더에 Application.mk작성 필수	Cygwin/home/(사용자계정)/.bashrc파일 마지막에 Export ANDROID_NDK_ROOT=/home/(사용자 계정)/(팩키지버전) 추가	
make APP= 명령 사용	ndk-build 명령 사용	
r3와 r4,r7의 차이는 ndk-build 명령문의 차이 r4,r7에서도 make APP 명령을 쓰기 위해서는 ndk폴더/apps/폴더/프로젝트 배치하고 폴더에 Application.mk작성 필수 ndk-build 명령문의 등장으로 ndk-build명령을 프로젝트 배치에 상관없이 사용가능		
		Application.mk에 APP_PLATFORM := android-7 추가해야 make APP명령 사용가능

<Table 1> Summary of differences between NDK versions

안 NDK에 대해 다루는 국내 서적은 시중에 원서 말고는 몇 권 없었으며, 인터넷에서조차도 유용한 정보를 찾아보기 힘들었다. 하지만 프로젝트 끝날 무렵부터 NDK를 다룬 국내 서적이 출판된 것으로 보아 NDK에 대한 개발자들의 관심도가 비로소 증가하고 있음을 알 수 있다.

<표 1>은 NDK의 각 버전별 차이점을 요약하고 있다. 가장 문제가 되는 차이는 ndk-build 명령의 유무로, 버전 r4부터 등장한 이 명령은 기존에 사용한 make 명령을 대체한 것이다. 버전 r4나 r7에서 기존에 쓰던 make 명령을 쓰려면 프로젝트 위치를 'ndk폴더/apps/폴더/프로젝트'로 하고 폴더에 Application.mk 파일을 반드시 만들어 두어야 한다. 또한 버전 r7에서는 Application.mk 파일에 'APP_PLATFORM := android-7' 디렉티브를 추가해야만 make 명령을 사용할 수 있다[3, 4, 5].

이처럼 NDK는 그 활용에 대한 가이드가 아직 많이 부족한 단계지만 향후 안드로이드 앱 개발에서 중심 작용을 할 수도 있다는 판단이나 올 정도로 사람들의 관심을 많이 받고 있는 추세이다.

2.2 안드로이드 NDK 동향

2.2.1 안드로이드 NDK

안드로이드 앱 개발 시 C나 C++과 같은 네이티브 코드(native code)를 사용할 수 있게 해주는 개발 도구이다. 메모리가 많이 필요한 경우나 안드로이드 앱에서 C/C++ 라이브러리를 사용하고 싶을 때 사용한다. <표 2>는 안드로이드 NDK와 안드로이드 SDK를 비교하고 있다[6]. NDK는 간단히 말해 자바 가상머신 기반에서 SDK가 기본적으로 지원하지 않은 기능들을 안

<표 2> NDK와 SDK의 차이점 요약

NDK	SDK
Native Development Kit	Software Development Kit
C, C++ 기반	Java 기반
OpenGL ES 2.0 지원	OpenGL ES 1.x 지원
리눅스 커널단에서 게임이나 장비 연동 등 특화된 API들을 가짐	Dalvik virtual machine 위에서 주로 앱이나 UI에 특화된 API들 가짐
Native인 C로 구현 되었기 때문에 속도가 빠르다	JAVA로 구현 되었기 때문에 NDK에 비해 속도가 느리다.

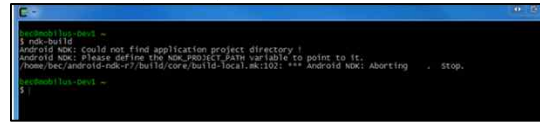
<Table 2> Summary of differences between android NDK and SDK

드로이드 커널 단에 추가 개발하고 이를 상위 단에서 쓸 수 있게 해준다. 따라서 영상처리 및 게임과 같이 속도에 민감한 모듈들을 개발하기에 적합한 안드로이드 시스템 프로그래밍 도구라 할 수 있다.

2.2.2 씨그윈(cygwin)

윈도에서 유닉스 계열 명령라인 환경을 쓰도록 도와주는 콘솔 유틸리티들의 모음이 씨그윈(cygwin) 이다[7]. 시그너스 솔루션스에 의해 개발된 자유 소프트웨어로 다양한 버전의 윈도에서 유닉스 계열 명령어들을 사용할 수 있게 해준다. GNU GPL로 배포되어 자유롭게 사용할 수 있다. (그림 1)은 씨그윈 실행화면을 보이고 있다.

(그림 1) 씨그윈 실행화면



(Figure 1) Running the cygwin

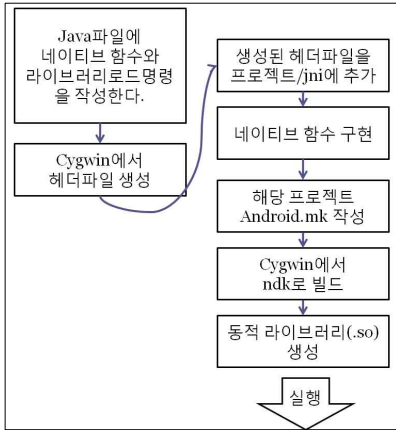
2.2.3 JNI

자바와 C/C++ 모듈간의 인터페이스를 가능하게 해주는 역할을 한다. 안드로이드 프레임 워크에서 C/C++과 자바 레이어가 유기적으로 동작하게 만들려면 자바 레이어와 C/C++레이어를 상호 연결해주는 매개체가 필요한데, 그 작업을 JNI가 담당하고 있다. JNI의 필요성은 다음과 같다[8].

- 빠른 처리 속도를 요구하는 루틴 작성: 자바는 네이티브 코드(C/C++ 등)에 비해 느리다. 따라서 빠른 처리 속도를 요하는 부분은 C/C++로 작성하고 이를 JNI를 통해 자바에서 호출하는 방법으로 속도 향상을 꾀할 수 있다.
- 하드웨어 제어: 하드웨어 제어 코드를 C로 작성한 다음 JNI를 통해 자바 레이어와 연결하면 자바에서도 하드웨어 제어가 가능하다.
- 기존 C/C++ 프로그램의 재사용: 이미 기존 코드가 대부분 C/C++로 작성돼 있다면 굳이 자바로 동일한 코드를 다시 작성하기 보다는 JNI를 통해 기존 코드를 활용할 수 있다.

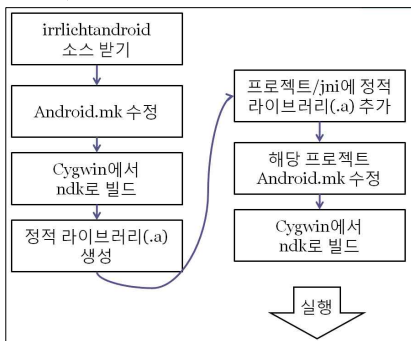
(그림 2)는 안드로이드 NDK 프로그래밍 시 C/C++로 작성된 동적 라이브러리 로드과정을 보이고 있고 (그림 3)은 정적 라이브러리 로드과정을 보이고 있다[9].

(그림 2) NDK 동적 라이브러리 로드과정



(Figure 2) Steps for NDK dynamic library loading

(그림 3) NDK 정적 라이브러리 로드과정

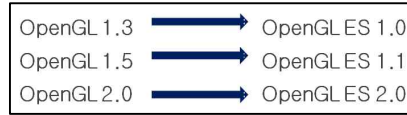


(Figure 3) Steps for NDK static library loading

2.2.4 OpenGL과 OpenGL/ES

OpenGL은 1992년 그래픽스사에서 만든 2차원 및 3차원 그래픽스 표준 API 규격으로, 다양한 플랫폼 상의 다양한 언어에서 사용가능하다. OpenGL/ES[10]는 OpenGL[11]의 서브셋으로, 휴대전화나 PDA 등과 같은 임베디드 단말을 위한 API이다. 여기서 ES는 Embedded System의 약자이다. OpenGL과 OpenGL/ES 간에는 다소 버전간 불일치가 있는데, (그림 4)는 OpenGL 과 OpenGL/ES 각 버전 관계를 보이고 있다.

(그림 4) OpenGL과 OpenGL/ES의 버전 관계



(Figure 4) Version Relations between OpenGL and OpenGL/ES

하지만 OpenGL/ES는 OpenGL의 서브셋으로 몇몇 기능들이 제공되지 않는데, OpenGL/ES의 제한점을 요약하면 다음과 같다[12].

- 지원하는 타입 수가 적다. 예를 들면, GLdouble 타입 등을 지원하지 않는다.
- 즉시모드를 지원하지 않는다. 예를 들면, glBegin() ~ glEnd() 블록 안에서 그리기 함수 호출이 불가하고, 정점 배열만 사용 가능하다.
- 사각형을 그리는 glRect 함수가 제외됨에 따라 모든 물체를 삼각형으로 구성해야 한다.
- 색상 인덱스 모드는 지원하지 않는다. 이는 조명이나 텍스처 맵핑 등에 제약이 많다는 것을 의미한다.
- 폴리곤모드, 피드백, 선택기능, 누적버퍼, 출력목록, 속성 저장기능 등을 지원하지 않는다.
- 텍스처 맵핑은 2D만 지원된다.
- 조명은 앞뒷면이 반드시 동일해야 한다. 재질 모드는 GL_ambient_and_diffuse만 지원한다.

2.2.5 Irrlicht

Irrlicht[2]는 C++로 만들어진 높은 고품질 다중 플랫폼 3차원 그래픽 엔진으로, 게임이나 과학적인 시각에 필요한 2D/3D 그래픽 함수들을 지원하는 것이 특징이다. Irrlicht의 장점을 요약하면 다음과 같다[13].

- 순수 C++로 개발되었고 객체지향적이다.
- Direct 3D와 OpenGL을 이용한 고수준 실시간 3D 렌더링이 가능하다.
- 빠르고 쉬운 충돌검사가 가능하다.
- 오픈 소스이며 소스 수준에서 버그를 고치거나 바꿀 수 있다.
- 플랫폼에 독립적이며, 지원하는 운영체제는 윈도, 리눅스, OS X 등이 있다.

- 픽셀, 정점, 도형 셰이더를 지원하는 확장 가능한 풍부한 텍스처 라이브러리를 내장하고 있다.
- 두 개의 플랫폼과 드라이버로 독립적으로 빠른 소프트웨어 렌더링을 포함한다.
- 일반적인 메시 파일 포맷을 직접 들여온다. (Maya의 obj, 3D, Studio의 3ds) 또 텍스처를 직접 들여올 수 있다(bmp,png,psd,jpg 등).

3. 구현환경 및 구현과정

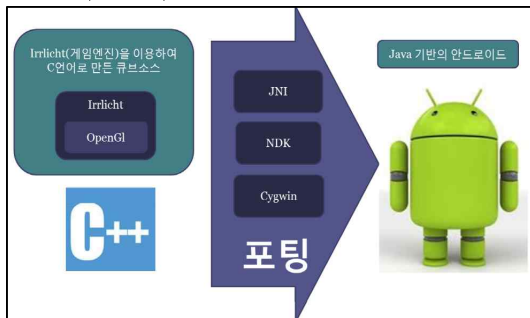
3.1 구현 환경

똑똑이 큐브의 개발 환경은 다음과 같다. 안드로이드 프로그래밍을 위해 JDK 1.6.0과 Android 2.2, Eclipse helios 혹은 indigo를 이용하였으며, 네이티브 언어 빌드 환경으로는 NDK r6b와 Cygwin 1.7.11을 사용하였다. 3D 그래픽 라이브러리로는 윈도우 상에서는 irrlicht 1.5.2를 사용하지만 안드로이드 상에서는 irrlichtandroid를 사용한다. C 언어 프로그래밍은 Microsoft Visual Studio 2005를 사용했으며, 그래픽 디자인 작업은 Adobe Illustrator CS5를 사용하였다. 테스트 기기로는 Galaxy M과 Galaxy Tab 10.1를 사용하였다.

3.2 구현 과정

똑똑이 큐브의 개발과정은 (그림 5)와 같다. OpenGL 기반의 게임엔진 irrlicht를 이용하여 C언어로 작성된 윈도우용 큐브 소스를 JNI, NDK,

(그림 5) 똑똑이 큐브의 개발 과정



(Figure 5) Development process for our android 3D cube application

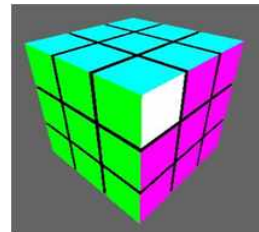
Cygwin을 통해 자바 기반의 안드로이드 응용에

서 사용할 수 있도록 한다.

3.3 큐브 자료구조 및 회전 방법

큐브를 구성하는 각 셀은 Node, Index, savecolor를 멤버로 하는 구조체로 표현된다. (그림 6)과 같이 3×3 큐브에서는 이런 셀이 총 27개가 있으며, <표 3>처럼 각 셀을 식별할 때 Index를 쓰고 자세한 상태 정보는 Node에 저장된다. savecolor는 완성도를 나타내는 알고리즘에서 사용된다. (그림 7)은 큐브의 회전 벡터들을 보이고 있다. 큐브를 회전할 때는 선택된 면과 커서 이동방향을 통해 x, y, -x, -y중 한 방향을 선택한다. 그 선택된 벡터와 노말 벡터를 이용해 회전벡터를 구한다[14].

(그림 6) 3×3 큐브의 셀 구조



(Figure 6) Cell structure of a 3×3 cube

<표 3> 큐브 셀 구조체의 구성

변수	역할
Node	한 셀의 각 면의 vertex, vector, id, color 등을 저장하는 구조체 포인터
Index	한 셀의 식별자
savecolor	해당 셀의 초기 color를 저장하는 배열

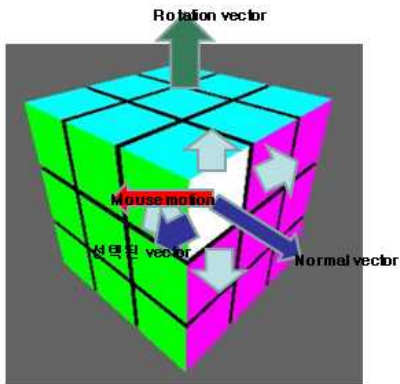
<Table 3> Data structure for a cell of cube

3.4 C++ 큐브 소스를 안드로이드로 이식하기

3.4.1 Irrlicht 라이브러리 빌드

Irrlicht 라이브러리는 include 폴더내의 170개 헤더파일과 정적 라이브러리, 그리고 동적 라이브러리(.so) 형태로 설치된다. Android.mk 파일에 동적 라이브러리로 만들고자하는 정적 라이브러리와 C 파일명을 추가하여 작성하고, 씨그

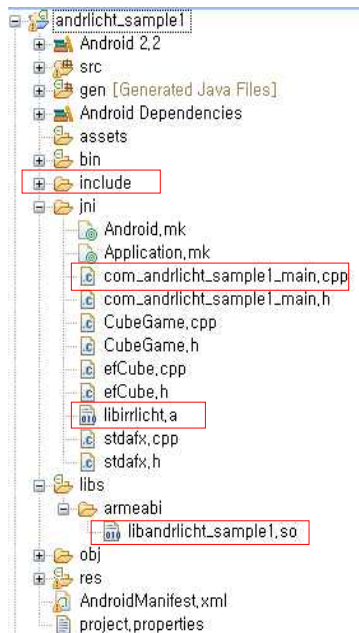
(그림 7) 큐브의 회전 벡터들



(Figure 7) Rotation vectors of a 3×3 cube cell

원에서 NDK로 빌드하면 동적 라이브러리가 libs 폴더와 함께 생성된다. (그림 8)은 빌드완료 후 이클립스에서 본 폴더 모습이다.

(그림 8) 빌드 완료 후 프로젝트 폴더 구성



(Figure 8) Contents of Project folder after completion of build

3.4.2 터치 이벤트 처리

윈도용 3D 큐브 소스에서 이벤트 처리를 제외한 다른 부분은 그대로 이식이 가능하지만 이벤트 처리 부분은 그대로 이식이 불가능하다. (그

림 9)는 윈도에서의 이벤트 처리 방식이고, (그림 10)은 이를 안드로이드에 맞게 수정한 방식을 보이고 있다.

(그림 9) Window에서의 이벤트 처리

```

if(클릭된 버튼이 오른쪽버튼인가?)
    큐브전체 회전
else{
    if(클릭된 버튼이 왼쪽버튼인가?){
        화면의 어느 부분에 충돌했는지 검사
        if(블록에 닿았는가?)
            해당블록을 흰색으로 표시
        }else{
            if(흰색으로 표시된 블록이 있는가?)
                회전방향을 벡터로 얻어냄
        }
    }
}
    
```

(Figure 9) Event handling flow in Windows

(그림 10) 안드로이드에서의 이벤트 처리

```

입력이 없을 때 변수=0
if(ACTION_DOWN인가?)
    화면의 어느 부분에 충돌했는지 검사
    썬이면 변수=1, 썬이 아니면 변수=2
else if(ACTION_UP인가?){
    변수=1일 때 회전방향을 벡터로 얻어냄
}
else if(ACTION_MOVE인가?)
    변수=2일 때 큐브전체 회전
    
```

(Figure 10) Event handling flow in Android

3.4.3 JNI 함수 구현

자바로 작성된 사용자 인터페이스 코드에서 입력받은 좌표값을 C로 된 큐브코드로 전달하기 위해 JNI 함수를 작성한다. (그림 8)에서 언급된 com_andrlicht_sample1_main.cpp 파일에 JNI 함수들이 있으며, 이 함수들을 통해 좌표값을 전달한다.

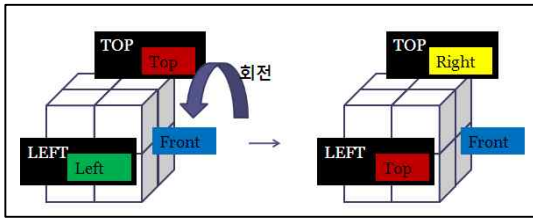
3.5 추가 기능 구현

3.5.1 큐브 완성 진척도 보여주기 기능

큐브의 각 셀은 3.3절에서 보인 구조체로 되어 있고, 이 구조체들은 mCube[n][n][n]배열에 각각 저장된다. mCube 배열을 컨테이너라고 생각하면, (그림 11)과 같이 mCube는 고정되어 있고 회전 시 블록의 구조체가 이동한다고 생각할 수 있다. 이 같은 점을 이용하여 (그림 12)와 같이 큐브가 어느 정도 완성되었는지를 퍼센트로 나타내는 알고리즘을 구현하였다.

3.5.2 페이스북 라이브러리 연결

(그림 11) 큐브 회전 시 자료구조 변화 예



(Figure 11) Example of a cell data position and direction changes after a cube rotation

(그림 12) 큐브 완성 진척도 계산 과정

```

초기화할 때, 배열의 초기 색을 저장한다.
if(큐브가 회전하여 배열의 정보가 변했는가){
    n*n*n개의 셀의 6개의 면을 초기 색과 비교
    if(초기 색과 일치하는가?)
        if(초기 색이 검정이 아니었는가?)
            일치하는 면 개수에 추가
        퍼센트를 구한다.
}
    
```

(Figure 12) Cube completion level estimation algorithm

이 기능은 게임을 하던 중 페이스북에 연결하여 현 게임 진행 상태를 올릴 수 있게 해준다. (그림 13)은 페이스북 라이브러리를 연결하는 과정을 보이고 있다[15].

(그림 13) 페이스북 라이브러리 연결 과정



(Figure 13) Linking the facebook library with our project

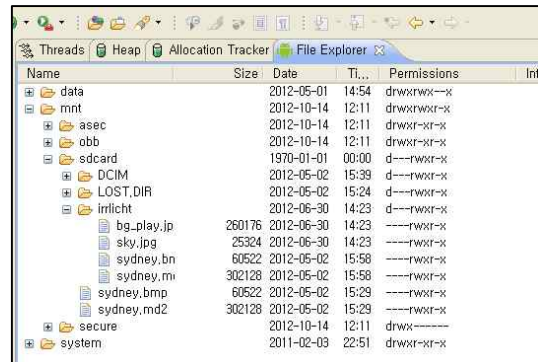
3.5.3 기타 추가 기능들

재미를 위해, 효과음/배경음/타이머/큐브면 이미지 바꾸기 등의 기능을 추가하였다. 안드로이드 프로그래밍과 JNI 함수 작성, C 프로그래밍을 복합적으로 사용하여 완성하였다.

3.6 이미지 제작

게임화면의 배경이미지는 C로 된 큐브코드에서 그린다. 화면에 맞게 자동으로 이미지 크기가 조절되는 자바와는 달리 C 레벨에서는 이미지 크기가 조절되지 않는다. 그래서 모바일 기기의 화면 크기에 맞추어 이미지를 각각 만들었다. 만든 이미지는 (그림 14)와 같이 DDMS를 통해 모바일 기기나 에뮬레이터의 SD 카드에 추가해 주어야 한다. DDMS(Dalvik Debug Monitor Server)는 ADT(android development Tool)와 함께 설치되는 디버깅 도구인데, 안드로이드 앱 개발에 필수적인 도구들을 모아놓은 통합 디버깅 프로그램이다[16]. DDMS 탭에서 sdcard 폴더에 irrlicht 폴더를 만들고, 그 폴더에 이미지를 넣어준다.

(그림 14) DDMS 실행 모습



(Figure 14) Running the DDMS

4. 실행 화면 및 평가

4.1 메인 화면

(그림 15)는 똑똑이 큐브의 메인화면이다. 2x2 ~ 4x4까지 원하는 크기의 큐브 모양을 클릭하면 게임화면으로 이동하고 큐브게임이 시작된다. (그림 15) 왼쪽 그림 하단의 물음표 버튼은 게임설명 화면으로, 톱니바퀴 버튼은 환경설정으로 각각 연결된다.

4.2 게임진행 화면

똑똑이 큐브의 게임 진행 화면은 (그림 16)과 같다. 화면을 터치하는 이벤트를 주어 사용자가 의도하는 방향으로 조각이 선택되고 회전하게

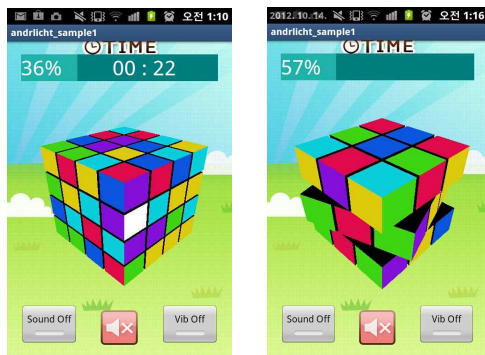
(그림 15) 메인 화면



(Figure 15) Main view of our 3D cube app.

된다. (그림 16) 하단에 보이는 버튼은 왼쪽부터 순서대로 효과음, 배경음악, 진동효과이며 좌측 상단에 %수치는 완성도를 나타내고 그 우측 옆 숫자는 타이머이다. 효과음은 큐브 회전 시 큐브가 돌아가는 소리를 내고, 진동효과는 큐브 회전 시 진동을 발생시킨다. 완성도는 큐브가 완성된 정도를 비율로 나타내며, 타이머는 사용자가 화면을 터치한 순간부터 카운팅된다.

(그림 16) 게임 진행 화면



(Figure 16) Playing our 3D cube game

4.3 큐브면 색 바꾸기

환경설정에서는 ‘큐브면 바꾸기’ 기능을 제공한다. 기본, 파스텔, 원색, 탁한색 중에 사용자가 원하는 스타일을 선택할 수 있다. (그림 17)은 파스텔을 선택하고 게임을 진행하는 화면이다.

4.4 게임 설명 화면

(그림 18)은 갤러리 형식의 게임 설명 화면으

(그림 17) 큐브면 색 바꾸기



(Figure 17) Changing colors of cube faces

로 큐브 맞추기 힌트도 제공한다.

(그림 18) 게임설명 화면



(Figure 18) Game manual page

4.5 페이스북 연결 화면

환경설정에서 페이스북 접속이 가능하도록 하였으며(그림 19), 큐브 게임을 하던 중 현 상태를 올릴 수 있다.

(그림 19) 페이스북 연결 화면



(Figure 19) Moving to my facebook account

5. 결론

많은 모바일 게임이 등장하는 현재, 2D 게임은 다양하게 존재하고 있지만 이에 비해 3D게임은 많이 부족하다. 또 스마트폰의 활용도가 증가함에 따라 사용자들은 더 빠르고 더 좋은 기기와 앱을 요구하고 있으며, 향상되는 모바일 기기의 성능을 제대로 활용할 수 있는 프로그래밍 기법이 요구되고 있다. 이에 안드로이드 NDK가 새롭게 각광받고 있다. 따라서 우리는 NDK를 사용한 가벼우면서도 재미있게 즐길 수 있는 3D 큐브 게임을 구현하였다.

3D 모바일 게임 분야는 모바일 기기의 한계점과 NDK 프로그래밍에 대한 정보 부족, NDK 버전간의 문제 등 아직 해결되어야 할 점이 많다. 하지만 날로 향상되는 모바일 기기의 성능과 NDK에 대한 관심을 고려할 때 향후 모바일 3D 게임과 NDK 프로그래밍이 보편화 될 것으로 예상된다. NDK를 활용하여 보다 빠른 앱을 개발하고, C언어로 작성된 많은 소스코드를 안드로이드에서 재사용할 수 있다면 향후 스마트폰 사용자들의 생활은 모바일 3D게임의 일반화를 포함하여 더 편해질 것이다.

이에 본 연구는 재미있게 즐기며 두뇌 트레이닝에도 도움이 될 수 있는 게임을 제공하는 동시에 NDK 활용의 좋은 예가 될 것으로 기대된다.

References

- [1] Go Eun Lee, Jongwoo Lee, "Analyzing Effectiveness of the Web Application Development in Smartphone," Journal of Digital Contents Society, Vol.11, No.3, pp.323-330, Sep. 2010.
- [2] <http://irrlight.sourceforge.net/>
- [3] Vladimir Silva, Pro Android Games, APress, 2009.
- [4] <http://occamsrazr.net/tt/231>
- [5] <http://www.androidpub.com/1860163>
- [6] <http://dlwpals001.blog.me/60126715319>

- [7] <http://www.cygwin.com>
- [8] 송형주 외, 인사이트 안드로이드 : 안드로이드 프레임워크 동작 원리 분석, 위키북스, 서울, 2011.
- [9] Sylvain Ratabouil, Android NDK Beginner's Guide, PACKT Publishing, 2012.
- [10] <http://developer.android.com/guide/topics/graphics/opengl.html>
- [11] <http://www.opengl.org/>
- [12] <http://blog.naver.com/statistical?Redirect=Log&logNo=50116738539>
- [13] <http://blog.naver.com/khk6435?Redirect=Log&logNo=50134282707>
- [14] <http://hellzblog.egloos.com/2432173>
- [15] <http://jeehun.egloos.com/3998429>
- [16] <http://developer.android.com/tools/debugging/ddms.html>

고 은 별



2009 ~ 2012: 숙명여자대학교 멀티미디어학과 학사
2013 ~ 현재: 숙명여자대학교 멀티미디어학과 석사과정

관심분야 : 모바일 소프트웨어, 검색 알고리즘, 시스템 소프트웨어

김 녹 희



2009 ~ 2012: 숙명여자대학교 멀티미디어학과 학사

관심분야 : 모바일 소프트웨어



황 성 미

2009 ~ 2012: 숙명여자대학교 멀티
미디어학과 학사

관심분야 : 모바일 소프트웨어



이 중 우

1990년 : 서울대학교 컴퓨터공학과
(학사)

1992년 : 서울대학교 컴퓨터공학과
대학원(석사)

1996년 : 서울대학교 컴퓨터공학과
대학원(박사)

1996~1998년: 현대전자(주) 정보시스템사업본부 과장

1998~1999년: 현대정보기술(주) 책임연구원

1999~2002년: 한림대학교 정보통신공학부 조교수

2002~2003년: 광운대학교 컴퓨터공학부 조교수

2003~2004년: 아이닉스소프트(주) 개발이사

2004~현재 : 숙명여자대학교 멀티미디어학과 교수

2008년 : 뉴욕주립대 스토니브룩 Research Scholar

관심분야: Mobile System Software, Storage Systems, Computational Finance, Cluster Computing, Parallel and Distributed Operating Systems, Embedded System Software