

## 다중플랫폼을 위한 제너릭 사용자 인터페이스 모델

강효운 · 김진덕\*

### Generic User Interface Model for Multi-Platform

Hyo-woon Kang · Jin-deog Kim\*

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

#### 요 약

스마트폰 시장은 구글의 안드로이드와 애플의 iOS 그리고 윈도우폰 플랫폼 등으로 구성된다. 이들은 각기 다른 프로그램 코드방식을 사용하여 자신의 플랫폼에 맞는 애플리케이션 제작방법을 채택하고 있다. 플랫폼에 종속적인 프로그램 제작 시 이는 곧 비용증대 및 코드 활용도가 낮아지며, 플랫폼이 지원하는 형식의 차이는 사용자에게 플랫폼별 이질적인 화면을 제공한다. 본 논문에서는 다중환경에서 운용할 수 있는 플랫폼 비 종속적 제너릭 사용자 인터페이스 모델을 제안한다. 제안된 시스템은 다중플랫폼 상에서 동일 UI를 제공하여 사용자는 이질감 없는 사용이 가능하다. 각 플랫폼 및 해상도의 변경을 수용하는 모듈을 설계하며, 성능향상을 위한 화면전환 모듈을 구성하고, 각 플랫폼별 데이터 저장 방식의 차이를 해결하기 위한 DB연동 구성 모듈을 탑재하도록 한다. 구현결과는 어느 플랫폼에 서도 이질감 없는 수행이 가능함을 보여주었다.

#### ABSTRACT

Smart Phone market consists of several platforms such as Windows Phone, Apple iOS and Android of Google. Each platform uses different coding style respectively. Platform-dependent application results in high cost and low code reusability during development of application. It also provides heterogeneous display forms to users. In this paper, we propose a platform-independent generic interface model that could be operated in both the various smart phones and PC. Because the proposed model provides the same UI on multiple platforms, users can operate applications without awkwardness. The model includes a module which copes with the changes of the resolutions in various platforms and a module which switches scenes of each application for performance improvement. Moreover, a module which connects DB is also proposed in order to overcome the difference of way to store data in each platform. The implementation results show that the difference of display and operating manner in any hetero-platform is eliminated.

**키워드** : 제너릭 인터페이스, 다중플랫폼, 코드재활용

**Key word** : Generic interface, Multi platform, Code reused

접수일자 : 2014. 08. 09 심사완료일자 : 2014. 08. 29 게재확정일자 : 2014. 09. 11

\* **Corresponding Author** Jin-Deog Kim(E-mail:jdk@deu.ac.kr, Tel:+82-51-890-1745)

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.12.2926>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서론

현재 구글의 안드로이드와 애플의 iOS가 스마트폰 시장을 양분하고 있으며 MS의 window 모바일은 windows8 출시와 함께 통합 OS 형태로 시장 점유율을 올려가고 있다. 마켓에 등록되는 네이티브앱은 특정 스마트폰 OS 전용으로 개발되기 때문에 대상 OS에 대하여 안정적이며 높은 성능을 제공한다[1].

그러나 네이티브 앱의 문제점은 첫 번째 특정 OS에 종속적이라는 것이다. 따라서 스마트폰 OS별로 네이티브 앱을 모두 구현하여야 한다. 두 번째 다양한 스마트폰이 출시됨에 따라 단일 플랫폼에서도 OS의 버전별 파편화 문제를 가져오고 있다. 그림 1은 현재 시장 지배적인 안드로이드의 파편화에 대해 나타내고 있다. 단순 버전의 변화에도 파편화를 가져옴을 볼 수 있다[2].

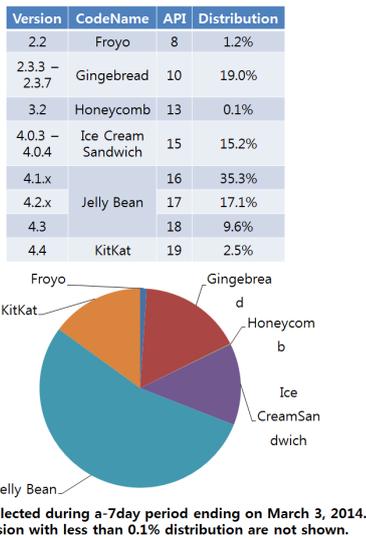


그림 1. 안드로이드 OS 점유율  
Fig. 1 Market Share of Android OS

이러한 문제를 해결하기 위해 현재 모바일 웹을 사용해 기존의 네이티브 앱을 대체하는 개발 방법이 증가하고 있다. 모바일 웹은 기존의 단점을 보완하며 네이티브 앱의 인터페이스를 운용할 수 있도록 구성하고 있다. 그러나 모바일 웹을 구성하는 HTML은 인터랙티브한 기능 부족 및 성능 저하라는 단점을 가지고 있다.

본 논문에서는 앞서 언급한 모바일 웹의 문제점과 네

이티브 앱의 OS 종속 문제 및 파편화에 따른 Device 종속 문제를 해결하기 위하여 제너릭 사용자 인터페이스 모델을 제안한다. 제안된 시스템은 플랫폼 및 해상도에 구애를 받지 않기 위하여 다중해상도 모듈을 구성하도록 하며, 모바일 웹의 단점인 페이지 전환 대신 화면 전환 모듈을 구성하고, 네이티브 앱 및 웹의 데이터베이스 차이를 없애기 위한 DB연동 모듈을 설계 제작하도록 한다. 이를 위하여 cocos2D와 JavaScript를 활용한다. 제안한 모델을 바탕으로 구현된 애플리케이션은 스마트폰 및 PC에서 구동 시 이질감 없는 화면을 구성한다.

본 논문의 구성은 다음과 같다. 2장에서 네이티브 앱의 OS 종속 문제에 대하여 언급하고 cocos2D 엔진 및 특징에 대하여 언급한다. 3장에서는 이를 활용한 다중 플랫폼 지원 사용자 인터페이스에 설계에 대하여 설명하고, 4장에서는 설계를 토대로 구현된 모바일 웹의 결과를 제시한다. 마지막으로 5장에서 결론을 맺는다.

## II. 관련 연구

파편화 문제를 해결하고자 플랫폼 비종속적 모바일 웹이 나타났으며 최근에는 HTML5를 활용하여 모바일 웹의 구조를 가지고 네이티브 앱으로 구성된 하이브리드 앱 개발이 많아지고 있다. HTML5는 다양한 환경을 고려한 웹 사이트 개발의 대안이 될 수 있다.

구글은 HTML을 적극 지원하여 플래시의 구동이 필요 없는 HTML5기반의 유튜브 서비스를 제공한다[3]. HTML5는 편리하며 구현도 매우 간단하지만 한 페이지는 여러 화면구성의 불편함을 가지게 된다.

이후 HTML5기반의 하이브리드 플랫폼으로 출시된 것이 폰갯이다. 폰갯[4]은 모바일 크로스 플랫폼을 지원하는 오픈소스이며, 모바일 개발 플랫폼이다. 폰갯 라이브러리를 이용하여 패키징 하는 과정은 공통으로 적용되는 패키징이 아니라 스마트폰 플랫폼에 따라 폰갯 라이브러리를 개별적으로 패키징 하여야 하는 문제를 가지고 있다. 센차 터치[5]는 또 하나의 모바일 웹앱을 구성하는 방법이다. 센차터치는 방대하고 체계적인 API를 제공한다. 그러나 센차터치를 이용할 경우 초기 로딩에 많은 시간적 소모를 보이며 프로그래밍 요소가 강한 반면 인터페이스 커스텀에 불편함을 보이고 있다.

cocos2D[6, 7]는 모바일 게임을 개발할 때 사용하는

게임 개발 엔진으로 모바일 웹과 네이티브 앱의 UI/UX를 이질감 없이 구현하도록 지원한다. 그림 2는 cocos2D의 엔진의 계층도이다. cocos2D는 OpenGL ES를 기반으로 만들어진 엔진으로 초기에는 iOS 기반의 게임을 개발하기 위해 만들어졌다.

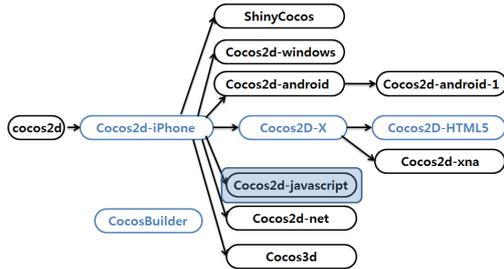


그림 2. 코코스2D 엔진 종류  
Fig. 2 Types of Cocos2D engine

복잡한 프로그램 코드의 경우 API 하부에 내재되어 있으므로 개발자는 편리하게 활용이 가능하다[8]. 본 논문에서는 제너릭 인터페이스 모델 설계를 통하여 네이티브앱과 모바일 웹의 동일 인터페이스 구현 모델을 제안한다. 이 모델은 모바일 웹, 네이티브 앱의 UI/UX와 동일하게 구현이 되며 코드 수정 없이 여러 OS에서 구동 가능하다.

### III. 제너릭 인터페이스 모델

#### 3.1. 시스템 구조

제안하는 모델은 어느 플랫폼에서나 동일한 환경의 UI를 제공하도록 한다. 이를 위하여 다중해상도를 지원하는 기법을 제안하고, 페이지 전환이 아닌 한 페이지에서 화면전환이 가능하도록 하는 기법을 제안한다. 그리고 저장방식 차이를 극복하도록 DB연동 모듈을 구성한다. 본 시스템은 그림 3과 같이 구성되며, HTML5의 인터페이스 단순화 및 페이지 전환 시 데이터 삭제의 문제를 보완하도록 본 논문에서 제안하는 인터페이스 모델을 적용한 교육용 앱의 예를 들어 설명한다.

모바일 웹앱의 전체적인 시스템 구조는 멀티 해상도 모듈 및 화면전환 모듈을 활용하여 사용자 인터페이스를 구성하고, DB연동 모듈을 이용하여 모바일 웹앱의 데이터 저장 방식을 구성하도록 한다.

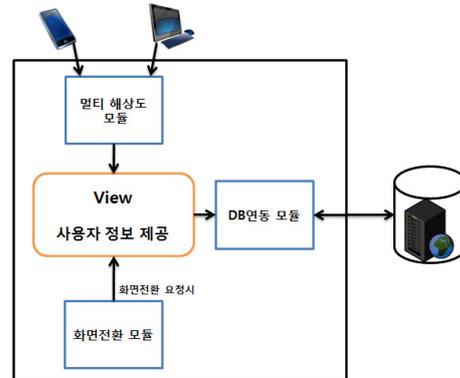


그림 3. 전체 시스템 구조도  
Fig. 3 System architecture

본 논문에서는 네이티브 앱의 UI와 연산처리과정을 동일하게 구성하였다. 모바일 웹앱은 특정 OS에 종속되지 않고 스마트폰, PC환경 등 비 종속적 OS의 특성을 바탕으로 어느 플랫폼에서도 운용이 가능하다는 장점을 가진다. 이러한 장점은 파편화의 단점을 해결하도록 한 번의 프로그램 구현으로 가능하게 한다.

본 시스템의 핵심은 사용자가 OS 및 화면 해상도에 관계없는 학습 환경을 제공 하며, 각 플랫폼에서 새로 패키징 하는 것이 아닌 모바일 웹상에서 직접 실행하여 어느 플랫폼에서나 구동이 가능하도록 한다.

설계된 시스템을 바탕으로 cocos2D와 JavaScript의 API를 활용하여 모바일 웹앱의 멀티 해상도 모듈과 화면 전환모듈 및 데이터 저장방식의 차이를 극복한 DB연동 모듈에 관해 각각 살펴보도록 한다.

#### 3.2. 멀티 해상도 모듈

다중플랫폼 및 동일 플랫폼에서도 각기 다른 해상도를 지원하는 OS의 경우 해상도의 차이가 발생한다.

특히 안드로이드 운영체제를 탑재한 스마트폰은 각기 다른 제조사가 제작하는 기기에 따른 이원적 해상도를 해결하기 위해서 해상도별 이미지의 제작과 UI를 새롭게 구성을 하는 문제를 가질 수밖에 없다. iOS 운영체제를 사용할 경우 안드로이드 플랫폼에서 제작된 네이티브 앱은 전혀 구동이 불가능한 문제를 가지고 있으며 PC환경에서도 실행 할 수 없다.

이러한 단점을 보완하기 위하여 모바일 웹앱이 출시되고 있다. 그러나 기존의 모바일 웹앱을 보면 HTML5 및 플래시 기반의 모바일 웹의 경우 해상도를 해결하기

에 부족함을 가지고 있었다. 또한 현재 플래시의 경우도 많이 축소화되고 있으며 플래시 또한 해상도의 문제를 모두 극복하거나 다중 플랫폼에서 운용하기 위해서는 각 플랫폼에 맞는 커백터가 필요하였다.

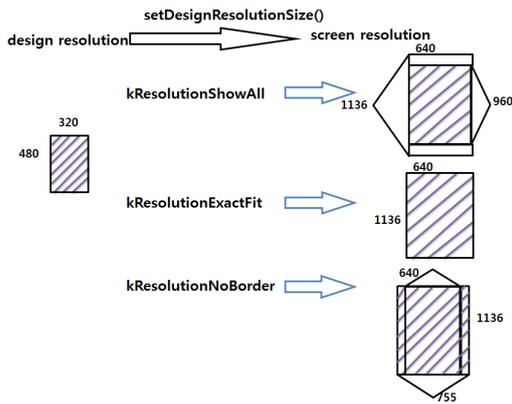


그림 4. 다중해상도 지원 방법  
Fig. 4 Method to support multi-resolution

그림 4는 다중해상도 지원 체계를 나타낸 것이다. 이는 해상도의 변경에 대응하기 위한 것으로 스케일링을 통하여 멀티 해상도 모듈을 구성하고 플랫폼 비종속적 웹앱을 구현하도록 지원한다.

```
bool AppDelegate::applicationDidFinishLaunching() {
    // initialize director
    auto director = Director::getInstance();
    auto glview = director->getOpenGLView();
    if(!glview) {
        glview = GLView::create("My App");
        director->setOpenGLView(glview)
    }
    setupResolutionPolicy(1080, 1920);
    director->setDisplayStats(true);
    director->setAnimationInterval(1.0 / 60);
    auto scene = ActionPage::createScene();
    director->runWithScene(scene);
    return true;
}
Size visibleSize = Director::getInstance()->getVisibleSize();
Sprite->setPosition(Vec2(visibleSize.width/2, visibleSize.height/2));
```

그림 5. 다중해상도 지원 코드  
Fig. 5 Code to support multi-resolution

그림 5는 다중 해상도를 지원하기 위하여 시작 시 화면 해상도의 크기를 가져오고 이를 바탕으로 스케일링을 실행 하는 과정 및 코드이다. 가장 작은 해상도에 맞

게 이미지를 제작 후 스케일링이 완료되면 이를 반환하고 해상도에 맞는 화면을 제공한다.

### 3.3. 화면 전환 모듈

모바일 웹은 일반적으로 여러 개의 페이지로 구성이 된다. 페이지 변경 시 잦은 통신이 이루어지며 페이지 전환 시 해당 페이지의 데이터는 사라지게 된다.

이러한 문제를 해결하기 위하여 화면 전환 모듈을 개발하여 HTML5의 화면전환의 문제를 보완한다. 새로운 페이지를 생성하는 것이 아닌 하나의 페이지에서 화면을 교환하거나 덮어쓰는 방법을 통하여 장면의 변화를 주는 것으로 문제를 해결하였다.

그림 6은 신 교환하기 방법 나타내는 것으로 각 장면에서 각각의 메인 모듈로의 장면 이동이 가능하며 화면이 메모리에 많이 쌓여 있지 않기 때문에 성능향상에도 높은 효과를 볼 수 있다.

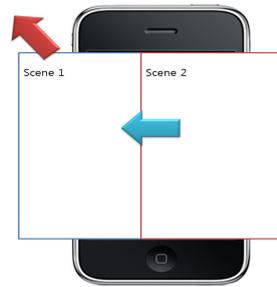


그림 6. 신 교환하기 방법  
Fig. 6 Method to change scenes

그림 7은 신(scene) 교환을 운용하여 화면간의 이동을 나타내는 코드 부분으로 교환 시 TransitionSlideInR을 이용하여 위에서 아래로 내려오며 이전 화면을 밀어내는 효과를 보이도록 한다.

```
callback: function () {
    var MenuScene = new Menu2()
    Director.sharedDirector.
    replaceScene(new nodes.TransitionSlideInR
    ({duration: 0.5, scene: MenuScene}))
```

그림 7. 신 밀어내기 코드  
Fig. 7 Code to push out scenes

그림 8은 신(scene) 덮어쓰기 방법을 나타내는 것으로 기존의 장면을 그대로 두고 그 위에 새로운 장면을

올려놓는 것이다.

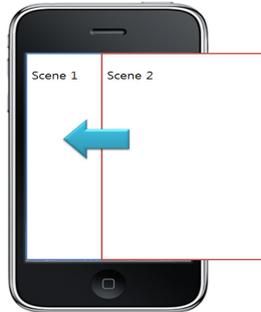


그림 8. 신 덮어쓰기 방법  
Fig. 8 Method to overwrite scenes

Scene 2 밑에 Scene1이 남아 있게 되는 것으로 메모리의 낭비를 불러올 수도 있지만 지속적인 화면의 이동 및 전환 시 데이터를 저장하여 잦은 데이터베이스 접근을 방지하여 데이터손실을 막는다.

본 모듈에 cocos2D의 공용 API를 활용하였으며 제안된 화면 전환 모듈은 UI별 독립적으로 운영하도록 한다. 이는 모든 모듈이 동시에 운용될 경우 앱의 속도저하를 가져올 수 있기 때문이다.

### 3.4. DB 연동 모듈

네이티브 앱을 운용하는 스마트 폰은 DB를 가질 수 있는 저장 공간이 있다. 하지만 모바일 웹앱의 경우 플랫폼에 맞는 데이터저장소가 없기 때문에 웹상의 DB저장소에 데이터는 저장되어야 한다. 따라서 본 논문에서는 JavaScript의 드라이버매니저를 활용 외부의 DB서버와 연동을 돕도록 한다.

```

final var Driver="jdbc:mysql://220.73.233.191/a/test";
final var User = "jtest";
final var Pass = "a12987";
var sql;
sql = "select * from question";
Try{ //예외처리
    Class.forName("com.mysql.jdbc.Driver");
}catch(ClassNotFoundException e) {
    out.print("where is your mysql ddbc driver?");
    e.printStackTrace();
    return;
}

conn = DriverManager.getConnection(Driver, User, Pass);
stmt = conn.createStatement();
rs = stmt.executeQuery(sql);
rs.next();
out.println(sql);
stmt.executeUpdate(sql);
    
```

그림 9. DB 접속 코드  
Fig. 9 Code to connect DB

그림 9는 모바일에서 MySQL에 접속하는 코드의 일부이다. DB접속모듈은 기존의 웹 파싱 방법을 탈피하여 직접적으로 DB에 접근하도록 한다. 이를 위하여 JDBC 라이브러리를 활용하였으며, 접속 후 데이터를 검색하고 다시 사용자에게 반환된 데이터는 저장 또는 업데이트를 진행 하도록 한다.

DB접속 모듈을 활용하여 직접적으로 DB에 접속하면 필수적으로 Date 타입을 맞추어야 한다. 이는 만일 다른 플랫폼에서 동시접속의 경우를 방지하는 기능을 위해서이다. DB의 Type설정이 완료된 후에는 JDBC를 이용해 연결을 하게 된다.

## IV. 플랫폼 비종속 모바일 웹앱 구현

앞서 제안한 제너릭 사용자 인터페이스 모델의 3가지 모듈인 멀티해상도 지원 모듈 및 화면전환 모듈 그리고 DB연동 모듈을 활용하여 모바일 웹앱을 구현하였다. 학습용 모바일 웹앱은 다중 플랫폼에서 구동이 가능하며 구현 결과는 다음과 같이 나타내도록 한다.

다음 그림10은 안드로이드 및 아이폰에서 구동되는 모바일 웹앱의 화면을 나타내는 것으로 두 플랫폼 모두 접속 시 동일한 구성의 화면을 가진다. 멀티 해상도 모듈이 적용되어 상이한 해상도에서도 같은 화면의 비율로 조정됨을 알 수 있다.



그림 10. 안드로이드(좌) 아이폰(우)의 주화면  
Fig. 10 Main Screen of andriod(left) & iPhone(right)

그림 11에서 모바일 웹앱은 하나의 프로그램을 통하여 다중 플랫폼에서 운용이 가능한 것을 보인다.



그림 11. 모바일 웹 앱(좌)과 네이티브 앱(우) 난이도 모드  
 Fig. 11 Mode to select difficulty in web app(left) & native app(right)

기존 모바일 웹 적용 시 기술의 부재로 네이티브 앱에 비하여 동작속도가 느린 문제점이 있었다. 이를 보완하기 위해 본 논문에서는 모바일 웹앱을 네이티브앱과 동일한 화면 구성을 하며, 이를 위해 난이도 선택에서 페이지 전환 기법의 신 교환하기 방식을 적용하여 메모리의 낭비를 줄여 다중 플랫폼에서도 네이티브앱과 동일한 속도의 움직임을 나타내었다.



그림 12. 모바일 웹 앱(좌)과 네이티브 앱(우) 학습 모드  
 Fig. 12 Study mode of web app(left) & native app(right)

그림 12는 문제 풀이 화면이다. 화면의 신 덮어 쓰기 방식은 메모리의 누적이 있을 수 있으나 문제의 앞, 뒤를 이동함에 있어서 문제 풀이가 완료 될 시점 까지 데이터를 담고 있으며 문제 풀이가 완료되면 데이터베이스

스에 접근하여 데이터를 저장하도록 한다. 신(scene) 교환 방식에 비해 신 덮어쓰기 방식은 데이터를 담고 있는 특징이 있어 모바일 웹앱의 대량의 데이터 손실을 막도록 한다. 또한 일반적으로 네이티브 앱의 경우 스마트폰에서만 사용됨으로 화면의 슬라이드 방식을 적용하는 것이 용이하나 구현된 모바일 웹앱의 경우 다중 플랫폼에서 활용하기 때문에 문제와 해답의 사이에 버튼을 적용하여 문제의 이동을 편리하게 할 수 있도록 하였다.

## V. 결론

본 논문은 제너릭 사용자 인터페이스 모델을 제안하였으며 이를 위하여 다중해상도 모듈, 화면전환 모듈, DB연동 모듈을 구성하고 제작하였다.

제안된 모듈을 이용하여 다중플랫폼에서 동일한 UI를 제공하고, 화면의 해상도에 구애 받지 않는 구성과 데이터의 손실을 줄이게 되었으며 제너릭 사용자 인터페이스 모델은 모바일 웹 기반으로 사용자의 편의 및 플랫폼 비 종속적 프로그램의 설계와 구현을 수행 하였다. 또한 개발에 있어 각 플랫폼별 개발이 아닌 하나의 코드를 이용하여 개발을 하였기 때문에 코드의 재활용도 매우 높아지게 되었다.

시스템 구현에 있어 일반적인 모바일 웹앱은 해상도의 문제와 화면전환이 되지 않아 페이지를 하게 될 경우 많은 데이터의 손실이 발생한 반면 제안된 모델은 해상도 및 화면전환에 매우 효과적이며 DB연동에 있어서도 데이터 낭비를 줄이게 되었다. JavaScript 및 HTML등을 활용하여 제작되는 일반적인 웹 기반 프로그램의 단점을 보완하도록 각각 상황에 맞는 변수 설정 및 DB연동에 활용하고, 사용자의 단말(PC, 스마트폰)에 의존적이지 않는 모든 해상도를 지원하는 멀티해상도와 플랫폼별 다른 UI 구성을 이질감 없는 UI를 보이는 장점을 가진다.

본 연구의 향후 과제는 모바일 웹앱을 운영할 경우 화면 전환 시 많은 데이터 사용이 발생하는데 이러한 데이터사용량의 최소화를 가질 수 있는 새로운 방법을 추진할 예정이다.

### 감사의 글

본 논문은 동의대학교 교내 학술 연구비(2014 AA250) 지원에 의한 결과물임을 밝힙니다.

### REFERENCES

- [1] G. E. Lee, and J. W. Lee, "Analyzing Effectiveness of the Web Application Development in Smartphone," *Journal of Digital Contents Society* vol. 11, no. 3, pp.232-330, Sep. 2010.
- [2] Simple version of the changes in the fragmented. Available: <http://OpenSignal.com>.
- [3] C. H. Nam, and C. G. Seo "A Study on the Implementation of Mobile Website Using HTML5," in *The Korea Society of Digital Policy & Management*, vol. 11, no. 1, pp. 165-172, Jen. 2013.
- [4] G. D. Song, G. H. Park, and S. H. Park "Recuperation Management System using Phoneyap and NFC," *KOREA SOCIETY OF COMPUTER INFORMATION*," vol. 21, no. 2, Sep. 2013.
- [5] R. Y. Jang, "Design and Implementation of Priority Analysis System Using AHP," Han. E. dissertation, Worcester Polytechnic Institute, Feb, 2013
- [6] Cocos2D-X Mobile UX/ UI. Available: [http://www.cocos2d-x.org/projects/cocos2d-x/wiki/Relationships\\_in\\_Cocos2d\\_X\\_Family](http://www.cocos2d-x.org/projects/cocos2d-x/wiki/Relationships_in_Cocos2d_X_Family)
- [7] Java Script Porting On Web Application. Available : <http://cocos2d-javascript.org/>
- [8] Cocos2D on Using OpenGL. Available : [http://www.cocos2d-x.org/wiki/Detailed\\_explanation\\_of\\_Cocos2d-x\\_Multi-resolution\\_adaptation?project\\_id=cocos2d-x](http://www.cocos2d-x.org/wiki/Detailed_explanation_of_Cocos2d-x_Multi-resolution_adaptation?project_id=cocos2d-x)



강효운(Hyo-Woon Kang)

2013년 2월 : 동의대학교 컴퓨터공학과(학사).  
2013년 3월 ~ 현재 동의대학교 컴퓨터응용공학과 석사과정  
※관심분야 : 데이터베이스, 모바일, GIS, 스마트 애플리케이션



김진덕(Jin-Deog Kim)

2000년 8월 : 부산대학교 컴퓨터공학과 졸업(공학박사)  
2001년 ~ 현재 : 동의대학교 컴퓨터공학과 교수  
※관심분야 : 데이터베이스, GIS, 모바일 시스템, LBS