

스마트폰 센서를 활용한 모바일 웹 어플리케이션 게임

임성호, 박상원

한국외국어대학교 정보통신공학과

e-mail: shlim2000@naver.com, swpark@hufs.ac.kr

Mobile web application game using smart phone sensor

Seongho Lim, Sangwon Park

Department of Information and Communication Engineering,
Hankuk University of Foreign Studies

요 약

닌텐도 Wii 출시 이후 가속센서를 활용한 체감형 게임기에 관한 관심이 급증하고 있다. 닌텐도의 경우 Wii 리모트에 가속센서를 탑재하여 사용자가 라켓이나 배트를 휘두르는 것과 같은 조작이 가능하게 한다. 스마트폰 또한 가속센서를 내장해 출시되고 있기 때문에 스마트폰의 가속센서를 활용해 게임을 개발할 수 있다. 하지만 스마트폰 플랫폼은 다양하기 때문에 한 플랫폼에서 개발을 완료하였다 하더라도 다른 플랫폼에서 동작하려면 대상 플랫폼에 맞게 프로그램을 제작성 해야 하는 단점이 있다. 해결책으로 웹 어플리케이션이 제안되어 왔다. 그러나 웹 어플리케이션은 자바스크립트 보안 정책 때문에 파일시스템, 센서 등 로컬 자원에 대한 접근이 불가능하다. 이러한 단점을 극복하고자 웹 어플리케이션이 센서를 조작하고 파일시스템에 접근할 수 있도록 지원하는 PhoneGap, BONDI 등의 솔루션이 개발되었다. 본 논문은 센서를 사용한 웹 게임의 구현을 통해 센서를 이용한 웹 게임 개발 방법을 제시한다.

1. 서론

최근 소니의 플레이스테이션3, 닌텐도 Wii 등의 체감형 게임기들이 출시됨에 따라 눈으로 보는 것뿐만 아니라, 실제로 몸으로 느끼고 즐기는 게임이 현실이 되고 있다. 세계적으로 게임 시장이 약한 성장세를 보이면서, 기존 게임 방식에 식상한 게임 유저들로부터 새로운 컨셉의 게임에 대한 요구가 급부상 하고 있다. 특히, 사용자와 컴퓨터를 연결하는 HCI(Human Computer Interface) 관점에서 보다 사실적이면서도 흥미로운 게임 인터페이스가 요구되고 있다. 이러한 게임 유저의 요구 사항에 발맞추어, 엑서게임(Exergame) 또는 엑서테인먼트(Exertainment)라는 신조어로 통칭되는 체감형 게임 시장이 개화하고 있다[1].

체감형 게임은 게임기에 내장된 카메라와 컴퓨터 비전 기술을 이용하여 게임 유저의 움직임을 파악하고 가속센서가 내장된 리모트 컨트롤러로부터 손동작을 인식한다. 소니의 경우 플레이스테이션 무브로, 닌텐도의 경우 Wii 리모트를 리모트 컨트롤러로 제공한다.

게임기가 기존의 단순한 게임패드 인터페이스로부터 현재처럼 다양한 센서를 이용한 인터페이스를 제공하는 것처럼 스마트폰 또한 단순한 전화기를 넘어 가속센서, 자기센서, 조도센서, 카메라 등이 탑재되어 출시되고 있다. 따라서 현재 게임 개발자가 스마트폰의 휴대성과 이동성을 활용한 체감형 게임을 개발할 수 있는 환경이 충분히 갖추어져 있다.

현재 스마트폰 플랫폼의 대표주자인 아이폰, 안드로이드, 블랙베리OS 등은 개발환경부터 사용 언어까지 모두 달라 어플리케이션의 호환성 보장이 불가능하다. 하지만 콘솔 게임에 비해 비교적 종류가 많은 모바일 플랫폼에서 게임은 호환성 또는 이식성이 좋아야 한다. 다양한 플랫폼에서 출시할 수 있어야 수익이 증대되기 때문이다. 따라서 다른 기종, 플랫폼에서 모두 구동될 수 있는 어플리케이션이 필요하다. 문제점의 대안으로 웹 브라우저 안에서 동작하는 웹 어플리케이션을 개발함으로써 이를 극복할 수 있다.

웹 어플리케이션이 구동되는 환경은 웹 브라우저 이다. 따라서 웹 브라우저가 설치되어 있는 환경에서는 모두 구동이 가능하다. 또한 HTML, 자바스크립트[2], CSS[3]를 사용하여 개발하므로 자연스럽게 MVC[4] 모델에 기반을 둔 개발이 가능해지며 이는 유지보수를 편리하게 해준다는 장점이 있다.

이러한 장점에도 불구하고 웹 어플리케이션은 치명적인 단점이 있다. 웹의 특성상 보안상의 이유로 파일 시스템과 스마트폰에 내장된 여러 센서 등에 접근할 수 없으며 'Same-origin policy'에 의해 타 도메인으로의 접근 또한 제한된다. 이를 극복하기 위한 대안은 첫 번째로 OMTP(Open Mobile Terminal Platform)의 BONDI[5]이다. OMTP는 브라우저 기반의 어플리케이션 혹은 위젯이 모바일 폰 기능을 보안적인 방법으로 접근하게 하는 모바일 웹 런타임 플랫폼인 BONDI를 발표하였다. BONDI의

경우 단말의 특정기능을 포팅된 형태로 제공하므로 별도의 웹 플랫폼이 설치되어야 한다. 따라서 별도의 웹 플랫폼을 제공해야한다는 단점이 있다.

두 번째로 Ajax 기술 전문기업인 Nitobi에서 오픈소스 프로젝트로 진행하여 개발한 PhoneGap[6]이다. PhoneGap은 기본 HTML을 바탕으로 하며 자바스크립트 엔진을 이용하여 기존의 HTML+자바스크립트만으로 모바일 환경의 기본 웹 브라우저에서 실행될 수 있다. 따라서 PhoneGap은 웹 애플리케이션의 한계를 별도의 플랫폼 없이 극복하므로 단말에서 바로 실행가능하며 각 플랫폼의 애플리케이션 스토어(Application Store)에서 바로 유통될 수 있다는 장점을 가지고 있다.

본 논문은 2장에서 관련 연구들을 소개하고, 3장에서 서론에서 제시한 웹 환경에서 스마트폰 센서를 활용한 게임을 구현하고 최종적으로 안드로이드 패키지 파일인 apk를 생성하는 과정을 소개한다. 4장에서는 결론 및 향후 개선 방안을 논한다.

2. 관련 연구

현재 출시된 스마트폰에는 많은 센서가 탑재되어 있다. 아이폰의 경우 자기센서, 근접센서, 가속센서, 조도센서, 중력센서가 내장되어 있으며 안드로이드의 대표적인 스마트폰인 넥서스원의 경우 중력센서를 제외한 아이폰의 모든 센서가 내장되어 있다. 자기센서는 자기장 또는 자력선의 크기와 방향을 측정하는 센서로 자기장의 영향으로 여러 가지 물질의 성질 등이 변하는 것을 이용하여 자기장을 측정한다. 스마트폰에서는 주로 디지털 나침반에 사용한다. 근접센서는 센서의 검출 면에 접근하는 물체, 혹은 근방에 존재하는 물체의 유무를 전자계의 힘을 이용하여 기계적 접촉 없이 검출하는 센서이며 가속센서는 이동하는 물체의 가속도나 충격의 세기를 측정하는데 사용한다. 특히 물체의 운동 상태를 상세하게 감지할 수 있으므로 활용 분야가 아주 넓고, 갖가지 용도로 사용되고 있다. 마지막으로 조도센서는 센서가 받는 빛의 세기를 그 면적에 비치는 광속으로 나타낸 양을 검출하는 센서로 스마트폰의 조명을 자동으로 점멸시키거나 그 밝기를 조절하는데 주로 사용한다.

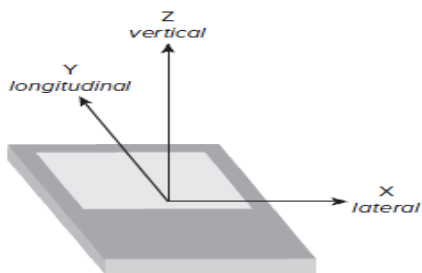


그림 1 가속센서의 가속도계[7]

시중의 체감형 게임기의 리모트 컨트롤러는 여러 센서들 중 가속센서를 이용한 센서가 많다. 가속센서는 그림 1의 x, y, z 세 가지 축의 가속도 값을 읽어온다.

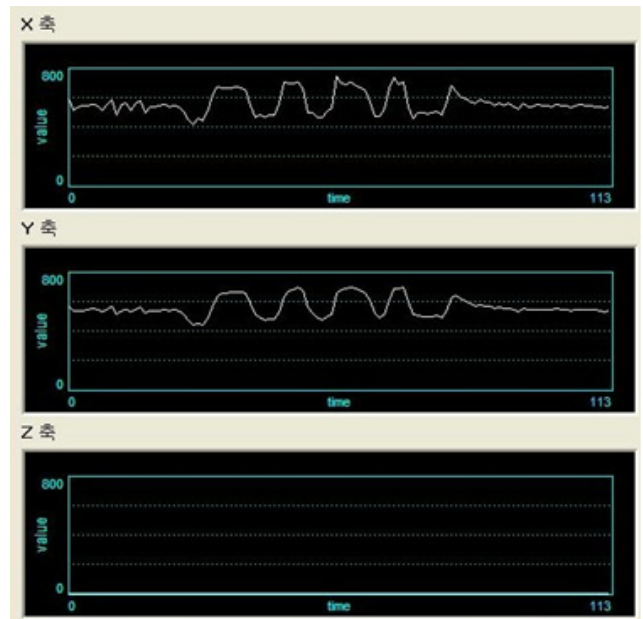


그림 2 시간에 따라 변하는 가속도 값[8]

가속센서가 탑재된 리모트 컨트롤러가 움직일 때 가속도 값은 시간에 따라 그림 2와 같이 변하는데 이 값을 게임 조작의 정보로 활용한다. 닌텐도 위의 테니스게임의 경우 리모트 컨트롤러를 라켓처럼 사용하여 실제로 휘둘렀을 때 가속센서로부터 읽어온 값을 토대로 공의 속도와 떨어질 위치를 결정하여 화면에 보여준다.

3장의 게임 구현에서는 이 가속센서를 사용한다. 플랫폼에 종속적이지 않은 게임 개발을 위해 게임을 웹 애플리케이션으로 구현할 것이다. 하지만 웹 애플리케이션은 샌드박스 안에서 동작하기 때문에 스마트폰의 로컬 자원인 센서에 접근할 수 없다. 따라서 웹 애플리케이션에서 센서를 사용하기 위해 Nitobi에서 개발한 오픈소스 하이브리드 애플리케이션 개발 도구인 PhoneGap을 사용한다. 웹 애플리케이션에서 로컬 자원 접근을 지원하기 위한 도구로는 대표적으로 OMTF의 BONDI와 Nitobi의 PhoneGap이 있다. BONDI의 경우 단말의 특정기능을 포팅된 형태로 제공하므로 별도의 웹 플랫폼이 설치되어야 한다. 따라서 별도의 구현된 웹 플랫폼이 있어야 하며 이를 애플리케이션 패키지에 포함해 제공해야한다는 단점이 있다. 반면 PhoneGap을 이용할 경우 기존의 웹 플랫폼을 그대로 이용하면서 장치와 웹 애플리케이션이 통신할 수 있는 방법을 제공한다. 따라서 패키지에 웹 플랫폼을 포함해 배포할 필요가 없으며 기존 플랫폼을 그대로 사용하므로 웹 애플리케이션 개발 외에 다른 비용이 들지 않는다. 또한 오픈소스이며 MIT 라이선스를 사용하므로 웹 애플리케이션의 제작과 배포에 제약이 없다는 장점이 있다. 하지만 미션

크리티컬한 기능이나 성능, 보안 및 로우 레벨 기술 적용이 필요한 어플리케이션에는 부적합하다는 단점이 있다.

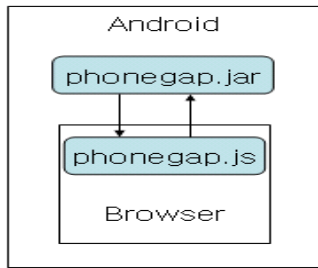


그림 3 PhoneGap의 동작

PhoneGap은 크로스 플랫폼 모바일 어플리케이션들(cross-platform mobile applications)을 위한 오픈 소스 개발 프레임워크이다. HTML과 Javascript를 이용하여 만들어진 어플리케이션이 모바일 기기의 자원에 접근하는 역할을 하며, 현재 아이폰, 안드로이드, 블랙베리, 윈도우 모바일, 심비안, 팜을 지원한다[9].

안드로이드의 경우 그림 1 과 같이 네이티브 자바코드에서 phonegap.jar의 형태와 자바스크립트 측에서 phonegap.js의 형태로 자원 접근을 지원한다.

PhoneGap을 이용한 어플리케이션은 웹 어플리케이션을 개발하듯 HTML과 자바스크립트로 개발하고 이를 네이티브 어플리케이션에서 불러와 화면에 띄워주는 방식으로 하이브리드 어플리케이션(Hybrid Application)이라고 한다.

표 1 PhoneGap이 지원하는 센서[7] O:지원 X:미지원

	아이폰	안드로이드	블랙베리	심비안	팜
위치정보	O	O	O	O	O
진동	O	O	O	O	O
가속센서	O	O	X	O	O
소리	O	O	O	O	O
주소록	O	O	O	O	X

PhoneGap은 표 1 와 같이 현재 WIFI에 기반을 둔 위치정보, 진동, 가속센서, 소리, 주소록 자원에 접근이 가능하며 그 외 나침반, 중력센서 등은 현재 지원하지 않는다.

3. 센서를 활용한 웹 어플리케이션 게임의 구현

이 장에서는 웹을 이용한 게임 어플리케이션을 구현하여 센서를 이용한 모바일 웹 개발에 대한 예제를 제시한다. 예제로 구현할 게임은 ‘돌피하기’로 플레이어가 하늘에서 떨어지는 돌을 피하는 간단한 게임이다. 가속센서를 이용하여 스마트폰을 좌, 우로 기울이는 방법으로 플레이어를 조작한다.

를 조작한다.

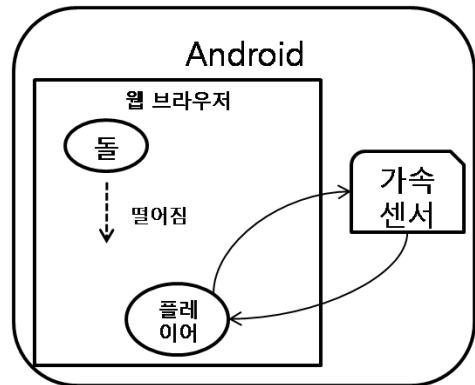


그림 4 프로그램 아키텍처(플레이어)

돌 피하기는 그림 4와 같은 구조로 표현할 수 있다. 그림에서 원안에 있는 요소는 HTML로 작성된 태그이고 점선인 화살표는 태그를 움직이는 자바스크립트라고 할 수 있다. 플레이어는 50 밀리 초 간격으로 가속센서의 x축 값을 읽어와서 부호가 양수이면 오른쪽 방향으로, 음수이면 왼쪽 방향으로 이동한다.

돌의 경우 50 밀리 초 간격으로 한 픽셀씩 아래로 이동하며 플레이어와 만나는지 혹은 바닥에 닿았는지 검사한다.

표 4 로컬 자원 접근의 예

```

Dol.prototype.stop = function() {
    clearInterval(this.interval);
    navigator.notification.vibrate(1);
}

Player.prototype.move = function() {
    function suc() {
        this.x = this.x + a.x;
        this.player = document.getElementById("player");
        player.style.left = this.x + "px";
    }
    navigator.accelerometer.wathAcceleration(suc, err, opt);
}
    
```

돌이 바닥에 닿았을 경우 표 4의 주요 소스 코드 중 navigator.notification.vibrate(1) 함수를 호출해 1초간 스마트폰을 진동시킨다. 가속센서의 조작은 표 4의 navigator.accelerometer.wathAcceleration()을 통해 기기의 중력 가속도 값을 측정한다. 측정된 중력가속도 값은 a 변수에 저장되고, a.x 또는 a.y 를 통해 x, y 값을 사용할 수 있다. 진동이나 가속센서를 사용하기 위해 스마트폰의 로컬 자원에 접근하게 되는데 PhoneGap은 phonegap.js와 phonegap.jar를 통해 이를 지원한다.

PhoneGap을 이용해 위의 소스를 빌드하면 자동적으로 안

드رويد 프로젝트를 생성하고 생성된 프로젝트를 빌드하면 안드로이드 패키지 파일이 만들어진다.



그림 5 PC 구동화면

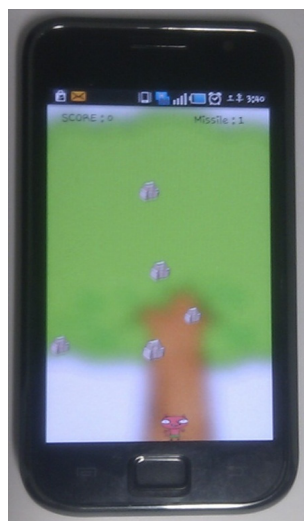


그림 6 갤럭시 구동화면

프로그램은 안드로이드 플랫폼인 갤럭시S, 아이폰, 아이패드, PC환경에서 실행하였다. 가속센서가 없는 PC환경에서는 플레이어의 조작을 위해 버튼을 추가하여야 한다.

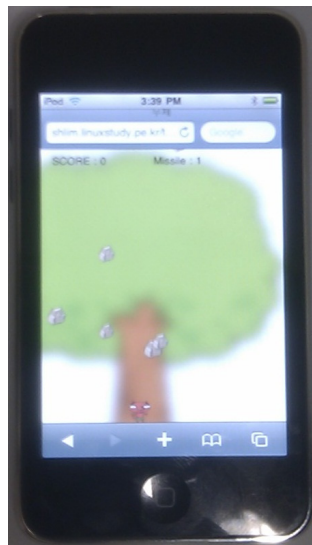


그림 7 아이폰 구동화면

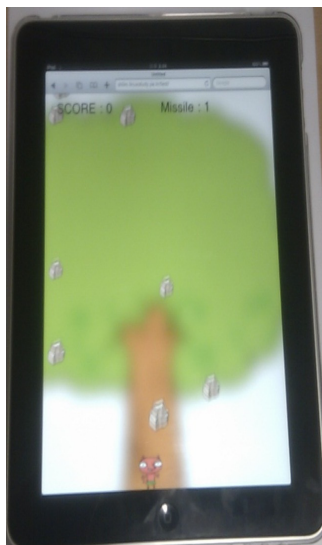


그림 8 아이패드 구동화면

4. 결론 및 향후 개선 방안

닌텐도 위 출시 이후 콘솔게임의 체감형 게임에 대한 관심이 급증하고 있다. 체감형 게임은 가속 센서를 많이 활용하는데 스마트폰 또한 가속 센서를 지원한다. 본 논문에서는 스마트폰의 가속센서를 이용해 체감형 게임을 구현하는데 플랫폼에 제한 없이 만들고자 웹 기술을 활용하였다.

게임을 웹 어플리케이션으로 구현하면 PC뿐만 아니라 아이폰, 안드로이드, 블랙베리, 심비안 등 다양한 플랫폼을 대상으로 배포할 수 있으며 플랫폼간의 포팅을 위한 추가

적인 노력도 필요하지 않다. 본 논문에서 소개한 웹 게임은 안드로이드를 타겟으로 개발을 시작하였지만 PC, 아이폰, 아이패드에서 테스트하기 위해 추가적으로 소요된 비용은 거의 없다. 또한 웹 어플리케이션이 센서를 제어할 수 있다는 사실을 바탕으로 스마트폰뿐만 아니라 스마트냉장고, 스마트 TV 등 OS가 탑재된 각종 가전제품간의 상호 호환이 가능한 어플리케이션을 개발할 수 있다. 또한 어플리케이션이 배포된 후 버그를 발견하여 유지보수 해야 할 상황이 생겼을 경우 네이티브 어플리케이션의 경우 각각 다른 기기에 포팅된 소스코드를 모두 수정하여 재배포 해야 하는 반면 웹 어플리케이션은 하나의 소스코드만을 수정하면 되므로 유지보수 측면에서도 효과적이다.

이러한 장점에도 불구하고 웹 어플리케이션에는 네이티브 어플리케이션에 비해 속도가 느리다는 단점이 있다. 또한 3장의 구현에서는 가속센서와 진동만을 이용한 게임을 개발하였지만 자기센서나 조도센서 등은 PhoneGap에는 구현되어있지 않다. 따라서 다양한 센서를 조작할 수 있는 API의 개발이 진행되어야 할 것이며, 브라우저 개발사들간의 표준 통일로 어플리케이션이 브라우저 종류별로 조금씩 다르게 동작하는 문제를 해결하는 작업이 선행되어야 할 것이다.

참고문헌

- [1] 이대현. "모션센서 기반 체감형 액션게임의 동향 및 개발 사례". 한국산업기술대학교, 한국콘텐츠학회
- [2] Shelley Powers. "Javascript for Web 2.0". O'REILLY
- [3] David Flanagan. "JavaScript: The Definitive Guide 5/E", O'REILLY
- [4] Gamma, Erich. "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley
- [5] BONDI 1.1. available at <http://bondi.omtp.org/1.1/>
- [6] Phonegap Docs. available at <http://docs.phonegap.com/>
- [7] Reto Meier. "Professional Android 2 Application Development", Wrox
- [8] 가속도센서 그래프. available at "<http://tskit.co.kr>"
- [9] Phonegap about. available at "<http://www.phonegap.com/about>"