

타이젠 프레임워크 가속도계 센서를 이용한 제스처 인식기능 확장

이성준, 신병석
인하대학교 컴퓨터공학과
e-mail : tjdwns5122@gmail.com, bsshin@inha.ac.kr

Extension of Gesture Recognition Function Using Tizen Framework Accelerometer Sensor

Sung-Jun Lee, Byeong-seok Shin
Dept. of Computer Science, In-ha University

요 약

타이젠은 오픈 소스 소프트웨어 플랫폼이다. 타이젠 SDK 를 통해 개발자는 타이젠 프레임워크 상에서 동작하는 애플리케이션을 손쉽게 구현할 수 있다. 그러나 타이젠 SDK 에서는 제스처를 인식하는 API 를 제공하지 않는다. 따라서 제스처를 이용한 앱을 개발하기 위해서는 앱 개발자가 센서 값을 분석해 제스처를 인식하는 기능을 구현해야 한다. 여기서는 제스처를 인식하는 기능을 기존 플랫폼에 추가해 앱 개발 시간을 단축한 구현사례를 소개한다.

1. 서론

타이젠 앱 개발자는 타이젠 프레임워크로부터 가속도계 센서 값을 받아올 수 있다. 가속도계 센서 값을 이용해 장치가 흔들렸는지 인식하는 기능을 추가할 수 있다. 그러나 보다 상세한 제스처를 인식하기 위해서 추가적인 작업이 요구된다.

타이젠 앱 개발자가 직접 이러한 기능을 구현할 수도 있으나, 앱 개발 시간이 길어지고 비효율적인 구현에 의해 사용자 경험(UX : User Experience)의 질이 낮아질 수 있다. 이러한 기능을 개별 앱 개발자가 구현하는 대신 타이젠 플랫폼이 제공하도록 수정하여 앱 개발기간을 단축하고, 타이젠 프레임워크에서 제스처를 인식하는 메커니즘을 제안한다.

2. 관련 연구

타이젠은 오픈 소스 소프트웨어 플랫폼이다. 스마트폰을 비롯한 휴대용 장치를 주력으로 하며, 태블릿, 넷북, 차량용 인포테인먼트 기기, 스마트 TV 등을 포함한 여러 종류의 기기를 원활하게 이용할 수 있도록 운영 체제와 애플리케이션을 제공한다. 타이젠은 SDK 를 제공하여 개발자가 타이젠 플랫폼 상에서 구동되는 애플리케이션을 쉽게 구현할 수 있도록 한다.

가속도계 센서는 장치의 속도 변화를 측정하는 것으로서 중력과 선형 가속도를 종합적으로 측정한다. 가속도계는 장치의 본체 프레임을 기준으로 한 3 개의 축에 대한 변위 벡터를 측정한다.

가속도계가 출력하는 값은 4 개로, Timestamp 값과 x, y, z 축에 해당하는 가속도 값이다. 여기서 x, y, z 축의 가속도에 해당하는 값들을 통해 기기가 어느 방향

으로 움직였는지 알 수 있고, 정적인 상태에서는 지속적으로 작용하는 중력가속도에 의해 장치의 방향을 알아낼 수 있다.

GBS(Git Build System)는 타이젠 패키지 개발을 지원하는 개발자 커맨드 라인 도구이다. 타이젠 네이티브 앱을 개발하려면 타이젠 SDK 가 있으면 되지만, 타이젠 플랫폼 자체를 빌드 하기 위해서는 GBS 가 있어야 한다. SDB(Smart Development Bridge)는 타이젠 SDK 에 포함된 장치 관리 툴이다. SDK 를 통해 연결된 장치에 여러 가지 기능을 수행할 수 있다. 여기서는 새로 빌드 한 타이젠 패키지 파일을 전송하고 설치하기 위해 사용하였다[1].

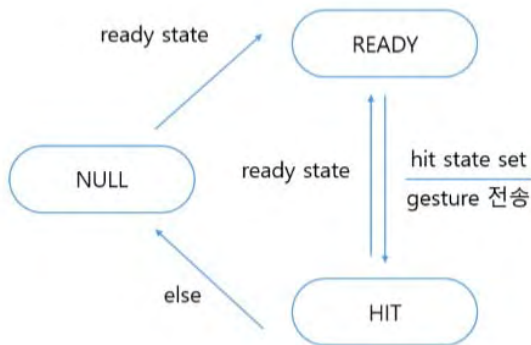
3. 제안하는 방법

타이젠에 있는 기존 제스처 기능은 사용자가 장치를 사용하며 하게 되는 자연스러운 동작만을 지원한다. 그러나 그러한 제스처와는 다른 인위적이고 간단한 제스처를 인식하기 위해서는 앱 개발자가 가속도계 센서 값을 받아와 분석하고 가공하는 과정이 필요하다. 여기서는 타이젠이 직접 그러한 기능을 제공하도록 프레임워크를 수정하는 것을 제안한다.

이후로는 가속도계 센서 값을 이용해 제스처를 인식하는 방법을 제시한다. 가속도계 센서 값을 이용하면 현재 장치의 방향을 알 수 있다. 현재 장치의 방향을 기록해두고 새로운 장치의 방향을 입력으로 받아 유효한 입력이 발생되었다면 제스처로 인식한다. 해당 과정을 상태전이도(state diagram)로 정리한다.

여기서는 상태를 3 가지로 구분하였다. NULL 상태는 장치가 어떠한 방향으로 변하더라도 제스처로 인

식하지 않는 상태를 의미한다. READY 상태는 아직 제스처가 인식되지 않았지만 다음 장치의 방향에 따라 유효한 제스처가 발생 가능한 상태를 의미한다. HIT 상태는 READY 상태에서 전이될 수 있는 상태로, 유효한 제스처가 인식되었다는 것을 의미한다. HIT 상태가 되면 타이젠 프레임워크가 콜백 함수를 통해 제스처가 발생했다는 의미의 신호를 전달한다.



(그림 1) state diagram

아래는 배열로 구현한 상태전이도의 소스 코드이다. 이중 배열의 입력으로 현재의 상태와 장치의 방향을 받아 다음 상태를 반환한다.

```

#define GS_R 0
#define GS_N 1
#define GS_H 2

int ngstate[3][7] = {
    {GS_R, GS_R, GS_H, GS_R, GS_H, GS_H, GS_H},
    {GS_N, GS_R, GS_N, GS_N, GS_N, GS_N, GS_N},
    {GS_N, GS_R, GS_N, GS_N, GS_N, GS_N, GS_N}};
    
```

(그림 2) 배열로 구현한 상태전이도

보다 구체적으로, 구현할 때에는 장치가 바닥에서 수직으로 서 있는 방향일 때에만 READY 상태가 될 수 있게 하였고, 장치가 바닥을 향해 기울어지는 방향으로 변할 때 HIT 상태가 되게 하였다. 또한 기울어지는 방향에 따라 각각 다른 제스처로 인식될 수 있게 구별하였다. 설계에 따라 보다 다양한 제스처 동작을 인식할 수 있다.

수정한 타이젠 프레임워크를 직접 장치에 적용하는 과정이 필요하다. 타이젠은 git 으로 관리되는 오픈 소스 프로젝트이기 때문에 개발자가 직접 소스 코드를 다운로드해 수정이 가능하다. 이렇게 다운로드한 소스 코드를 수정해 다시 빌드하고 장치에 설치할 수 있다.

타이젠은 rpm 패키지 형태로 배포되기 때문에 수정한 소스 코드를 rpm 형태로 빌드 해야 한다. 이를 위해 GBS 를 사용한다. GBS 를 통해 소스 코드를 rpm 패키지 파일로 빌드 할 수 있다. 빌드 할 때 모바일 장치의 cpu 를 알려주는 옵션을 추가해야 한다.

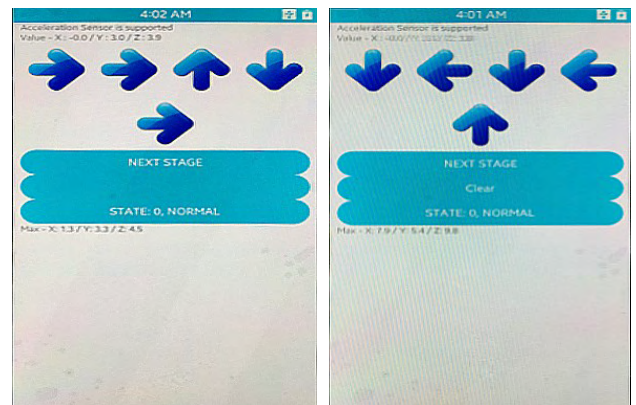
만들어진 rpm 패키지 파일을 모바일 장치에 설치하기 위해 SDB 를 사용한다. USB 로 타이젠을 설치할

모바일 장치를 연결하고, SDB 를 이용해 장치의 루트 권한을 얻은 뒤 rpm 파일을 push 한다. 그 후 rpm 패키지를 설치하고 장치를 재부팅하면 수정된 타이젠 프레임워크가 실행되게 된다[2].

4. 구현결과

실제로 수정한 플랫폼이 값을 전달받는가를 확인하기 위해서 타이젠 네이티브 앱을 개발해 실험하였다. 타이젠의 가속도계 센서가 반환하는 값은 본래 네개이다. 소스 코드를 수정해 새로 반환하게 된 다섯번째 값이 제스처에 해당한다. 이 다섯번째 변수인 제스처 값을 이용하는 앱을 제작해 제스처가 제대로 동작하는지 확인하였다.

실험용으로 만든 앱은 간이 DDR 게임 앱이다. 화면에 네개의 화살표를 표시해 해당하는 방향으로 모바일 장치를 기울이도록 지시하는 게임이다. 위에 있는 네개의 화살표는 따라 해야 할 제스처 방향을 의미하며, 아래에 있는 한 개의 화살표는 마지막 제스처 방향을 의미한다.



(그림 3) 간이 DDR 게임 화면

(그림 3)에서 앱을 사용한 결과 제스처 인식 기능이 제대로 동작하고 있음을 알 수 있다. 장치의 물리적 방향을 토대로 플랫폼이 제스처를 인식하고 그 정보를 네이티브 앱에 정확히 전달되고 있다. 이는 모든 화살표의 지시를 따른 결과 화면에 나타나는 Clear 라는 글자를 통해 앱에서도 옳은 값을 전달받았다는 사실을 통해 알 수 있었다.

5. 결론

타이젠이 오픈 소스라는 점을 이용해 프레임워크의 기능을 확장, 수정하여 재배포할 수 있다. 본 연구에서는 가속도계 센서의 값들을 가공해 새로운 값을 제공하였으며, 이를 바탕으로 타이젠 네이티브 앱 개발자가 더욱 쉽게 다양한 제스처 기능을 이용할 수 있도록 하였다.

참고문헌

[1] Tizen Association, <https://developer.tizen.org/ko>, 2012
 [2] Tizen Association, <https://source.tizen.org/>, 2012