

안드로이드 기반의 컬링 게임 개발

조 재 현* · 고 세 환* · 김 연 희** · 임 해 철***

Development of Curling Game based on Android Mobile Platform

Cho, Jae Hyeon · Ko, Se Hwan · Kim, Youn Hee · Lim, Hae Chull

〈Abstract〉

As expectations about success of smart phones that install Android, an open mobile platform which was developed by Google, are increasing, the development of applications based on the Android has attracted a lot of interest. Especially, game applications have been well received by users. In this paper, we develop an application based on the Android for curling which was adopted as a competitive sport in the Winter Olympic Games. Curling is a sport game played by two teams. The rules of the game are simple, but it is an intelligent one that requires various strategies for victory. In this paper, the curling game based on the Android provides functions, such as pitching, sweeping, collision, and foul handling, that are required in real curling game by using technologies, such as touch screen or accelerometer sensor.

Key Words : Mobile Game, Android Mobile Platform, Curling

I. 서론

휴대 단말기와 이동 통신 기술의 발달은 통화 목적의 휴대 전화를 넘어서 손 안의 작은 컴퓨터로 불리는 스마트폰의 개발을 이끌어 냈다. 해외에서는 현재 애플의 아이폰과 함께 최근 구글에서 제작한 모바일 플랫폼을 탑재한 안드로이드 폰이 크게 주목받고 있다. 국내에서도 표준 모바일 플랫폼으로 WIPI의 의무 사용이 폐지되면서 다른 모바일 플랫폼의 탑재가 가능한 스마트 폰의 출시가 본격화되고 있다. 이러한 스마트 폰의 실현에 가장 중요

한 역할을 담당하는 것이 휴대 단말기에서 동작하는 운영체제, 즉 모바일 플랫폼이라 할 수 있다. RIM의 블랙베리, 애플의 iOS, 마이크로소프트의 윈도우즈 모바일, 구글의 안드로이드 등과 같은 다양한 모바일 플랫폼들이 현재 스마트폰의 보급화를 지원하고 있다. 그 중 구글의 안드로이드는 현재 시장 점유율은 낮지만 가장 최선의 기술이며 리눅스 커널을 기반으로 하는 소스 코드를 공개하고 있기 때문에 국내외 많은 기업과 개발자들의 큰 관심을 받고 있고 향후 시장에서의 성장 가능성이 무엇보다 높다고 할 수 있다[1]. 그리고 구글에서는 안드로이드 환경에서 동작할 수 있는 애플리케이션을 개발한 후 다른 사람들과 쉽게 공유할 수 있도록 안드로이드 마켓을 운영하고 있기 때문에 안드로이드 폰에 적합한 다양한 애플리케이션의

* 홍익대학교 컴퓨터공학과

** 부천대학 e-비즈니스과 강의전담교수

*** 홍익대학교 컴퓨터공학과 교수(교신저자)

공급이 가능하다는 것도 안드로이드의 장점이다[1-2]. 이러한 이유로 스마트 폰의 출시와 함께 국내외 기업들과 개발자들이 안드로이드 폰을 위한 애플리케이션 개발에 큰 관심을 보이고 있다. 그리고 휴대 단말기의 보급과 함께 음악, 방송, 영상 등 다양한 분야의 모바일 콘텐츠 시장도 크게 성장하고 있지만 특히 일반 사용자들에게 많은 호응을 받고 있는 것이 게임 콘텐츠이다[3]. 따라서 안드로이드 환경에 기반한 새로운 게임 애플리케이션을 개발하게 되면 일반 사용자가 안드로이드 폰에 더욱 친숙함을 느끼게 되어 차세대 스마트 폰으로 각광받을 수 있는 배경을 마련할 수 있을 것으로 예상된다. 그래서 본 논문에서는 '얼음 위의 체스' 라고 불릴 만큼 높은 수준의 두뇌 싸움이 필요하지만 진행 규칙은 간단해서 일반 사용자들도 쉽게 즐길 수 있는 컬링(Curling) 게임을 안드로이드 플랫폼 기반에서 개발한 애플리케이션을 제안한다. 본 논문에서 개발한 컬링 게임 애플리케이션은 안드로이드 환경에서 터치스크린이나 중력 센서 기술을 이용해 사용자가 실제 컬링 게임을 진행하는 것과 같은 만족감과 흥미를 유도할 수 있어 플래시로 제작된 기존 컬링 게임 등과 차별화된 특징이 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 안드로이드 모바일 플랫폼과 컬링 게임에 대해 소개하고 3장에서는 본 논문에서 개발한 안드로이드 기반의 컬링 게임의 진행 방법과 개발 환경에 대해 소개하고 4장에서는 전체적인 시스템 설계 내용을 설명한다. 그리고 5장에서는 4장의 설계 내용을 바탕으로 개발된 컬링 게임의 실행 결과물을 제시하고 6장에서 결론을 맺는다.

II. 관련연구

본 논문에서 제안한 안드로이드 기반의 컬링 게임을 개발하기 위해서는 무엇보다 안드로이드 모바일 플랫폼에 대한 이해와 실제 컬링 게임에 대한 정확한 분석이 필요하다.

2.1 안드로이드 모바일 플랫폼

안드로이드는 모바일 기기를 위한 플랫폼으로 리눅스 커널 기반의 운영체제와 미들웨어, 주요 애플리케이션들을 포함하고 있으며 2007년 구글에서 발표하였다[1, 45]. 구글에서는 안드로이드 소스와 SDK(Software Development Kit)를 공개하여 개발자들이 안드로이드 환경에서 동작 가능한 다양한 애플리케이션을 개발할 수 있도록 하면서 안드로이드 마켓을 통해 누구나 개발한 애플리케이션을 자유롭게 공유할 수 있는 환경을 제공하고 있다. 안드로이드 애플리케이션들은 자바 언어로 작성되며 Dalvik이라는 가상 머신에서 실행된다.

안드로이드 플랫폼은 리눅스 커널, 라이브러리, 안드로이드 런타임, 애플리케이션 프레임워크, 애플리케이션의 다섯 계층으로 구성되어 있다[4]. 리눅스 커널 계층은 보안, 메모리 관리, 프로세스 관리, 네트워크 스택, 드라이버 모델 등의 핵심 서비스를 제공하는 역할을 담당한다. 라이브러리 계층은 안드로이드 시스템의 여러 컴포넌트에서 사용되는 라이브러리들을 제공하는 역할을 담당한다. 안드로이드 런타임 계층은 애플리케이션의 실행을 위해 필요한 Dalvik 가상 머신과 자바 언어의 핵심 라이브러리 기능들을 제공하는 역할을 담당한다. 애플리케이션 프레임워크 계층은 애플리케이션 개발에 필요한 기본적인 틀, 즉 프레임워크를 제공하는 역할을 담당한다. 프레임워크에는 뷰, 데이터 공유, 리소스 관리 등을 위한 다양한 클래스와 메서드들이 포함되어 있다. 애플리케이션 계층은 이메일 클라이언트, SMS, 지도, 브라우저 등과 같은 핵심 애플리케이션을 제공한다.

이와 같이 안드로이드 모바일 플랫폼은 잘 구성된 계층 구조를 통해 애플리케이션 개발에 필요한 방대한 양의 프레임워크와 라이브러리를 제공한다. 그리고 개방형인 리눅스 커널이 가지고 있는 장점을 그대로 제공하기 때문에 안드로이드 플랫폼의 보편화가 빠르게 진행될 것이라고 예상할 수 있다.

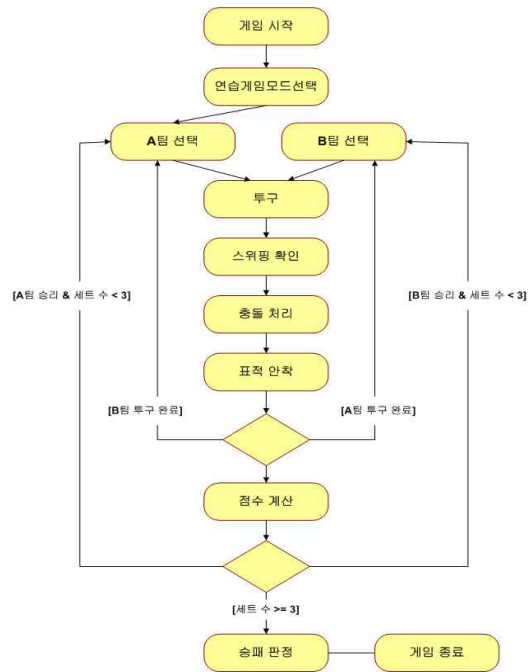
2.2 컬링

컬링은 두 팀이 얼음판에서 스톤(stone)이라고 불리는 둥글고 납작한 돌을 미끄러지게 하여 하우스(house)라 불리는 상대팀의 표적판 안에 넣어 획득한 점수를 겨루는 경기이다. 스코틀랜드에서 유래된 것으로 1998년 동계 올림픽 대회부터 정식 종목으로 채택되었다.

하우스는 각각 반지름이 다른 4개의 원으로 구성되어 있으며 가장 안쪽의 원을 티(tee)라고 한다. 한 게임은 10 세트에 걸쳐서 진행된다. 각 세트에서 4명으로 구성된 한 팀은 한 선수 당 2개씩 총 16개의 스톤을 번갈아가며 상대팀 하우스에 투구한다. 상대팀보다 티에 더 가깝게 보낸 스톤만 1점씩 인정되고 결과적으로 상대팀보다 티에 가까이 보낸 스톤의 수가 많으면 그 세트를 승리하게 된다. 첫 번째 세트에서는 동전을 던져서 공격 순서를 결정하고 두 번째 세트부터는 승리한 팀이 먼저 공격하게 된다. 양 팀이 모두 점수를 내지 못한 경우에는 다음 세트까지 공격 순서가 그대로 유지된다. 스톤의 투구와 더불어 스위핑(sweeping)은 컬링에 있어서 득점에 영향을 끼치는 중요한 작업이다. 얼음판과 스톤의 마찰력을 줄이기 위해 경기 시작 전 페블(pebble)이라고 하는 얼음 알갱이를 뿌려두는데 스위핑은 투구한 스톤이 앞으로 이동하는 동안 빗자루 모양의 솔을 이용하여 얼음판을 닦아서 스톤의 진행 방향과 속도를 조절해 목표한 지점에 최대한 가깝게 멈추도록 하는 작업이다. 컬링 게임은 스톤의 위치를 결정하고 진행 방향을 선택하는데 어떤 전략을 사용할 것인지 많은 고민을 해야 승리할 수 있기 때문에 '얼음 위의 체스'라고 불리기도 하지만 경기 규칙은 간단해서 연령에 관계없이 누구나 즐길 수 있는 게임이다.

III. 게임의 흐름과 개발 환경

전체적인 게임의 흐름은 <그림 1>에서 확인할 수 있다.



<그림 1> 게임 시스템의 전체적인 흐름

게임을 구동하면 게임모드 중 연습게임 모드를 선택한 후 게임을 시작할 수 있다. A팀과 B팀으로 나누어 게임을 진행하고 A팀이 먼저 공격한다. 각 팀 당 4개의 스톤을 번갈아 투구하고 모든 선수가 투구를 완료하면 1세트가 끝나게 된다. 투구된 스톤이 투구 라인과 표적판 사이에 위치한 중앙 라인을 넘어서지 못하면 그 즉시 얼음판 위에서 사라지게 된다. 투구된 스톤 중에서 표적판이 위치한 화면을 벗어나지 않고 안착한 것에 대해서만 점수로 계산된다. 표적판 중심에서 가장 가까이 위치한 스톤의 개수를 가지고 점수를 결정한다. 실제 컬링 게임에서는 한 게임이 10 세트로 구성되지만 모바일 게임으로 진행하기에는 게임 시간이 너무 길어질 수 있기 때문에 본 논문에서 개발한 모바일 컬링 게임에서는 3세트로 구성한다.

본 논문에서 제안한 모바일 컬링 게임은 안드로이드 플랫폼 기반에서 동작이 가능하도록 Android 1.1 SDK에서 제공하는 여러 도구들과 API를 이용하여 Eclipse 3.4.2에서 자바 언어로 구현하였다.

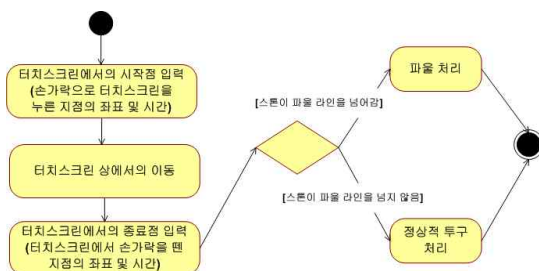
IV. 모바일 컬링 게임 설계

본 논문에서 개발한 모바일 컬링 게임은 투구, 스윙핑, 충돌, 점수 계산을 위한 4개의 주요 모듈로 구성되어 있다.

4.1 투구

투구는 스톤을 얼음판 위에서 미끄러지게 하는 동작으로 터치스크린 기능을 이용한다. 사용자의 손가락과 터치스크린이 처음 접한 위치와 손가락이 터치스크린에서 떨어지는 순간의 위치 사이에서 X, Y 좌표의 변화를 계산하여 스톤이 움직이게 될 방향을 결정하고 이동 거리를 고려해서 힘의 세기를 결정한다. <그림 2>는 투구 과정의 흐름을 설명한다.

투구 후 스톤이 움직이게 될 방향과 힘의 세기가 결정되면 실제 컬링 게임에서처럼 스톤과 얼음판 사이의 마찰력 때문에 스톤의 속력이 점차 감소하게 된다. 이것은 단위 시간당 이동 거리의 감소를 통해 표현하는데 예를 들어 0.1초 단위로 스톤의 움직임이 스크린 상에 표시된다면 0.1초마다 이동하는 거리가 짧아지도록 한다. 얼음판의 초기 마찰력은 0.02로 하고 이후 스윙핑 동작이 발생하는 상황에 맞게 마찰력에 변화를 주어 현실적인 컬링 게임이 되도록 하였다.



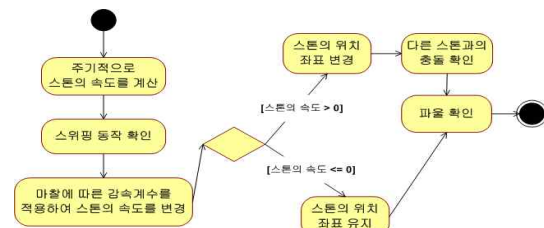
<그림 2> 투구 과정

터치스크린을 이용하여 투구를 끝낸 후 스톤의 진행 방향 벡터를 설정하는 모듈의 주요 부분은 다음과 같다.

```

// 터치스크린을 이용하여 스톤을 투구하는 과정을 처리하는 모듈
boolean doTouchEvent(MotionEvent event){
    if(투구를 시작하기 위해 터치스크린을 손가락으로 누른 경우){
        if(투구할 스톤이 남아 있는 경우){
            // 스톤의 투구 시작 위치 결정
            stoneSet[i].setLocation(event.getX(), event.getY());
        }
    }
    else if(투구를 끝내기 위해 터치스크린에서 손가락을 떼는 경우){
        // 아직 투구가 완료되지 않은 스톤 검색
        // 스톤의 진행 방향 벡터를 설정
        // 단, 전체 화면을 사용하기 위해 방향 벡터의 값을 작게 적용
        stoneSet[i].setVector((event.getX()-stoneSet[i].getX())/30.0,
            (event.getY()-stoneSet[i].getY())/ 30.0);
        // 투구 완료된 스톤의 개수를 증가시키고 설정된 방향 벡터에
        // 따라 스톤의 이동을 처리
    }
}
    
```

<그림 3>은 투구 동작 후 스톤이 움직이는 과정을 설명한다.



<그림 3>투구 후 스톤이 이동하는 과정

투구 동작이 마무리 된 후 스톤이 중앙 라인을 넘어가지 못하거나 표적판을 벗어난 경우에는 투구된 스톤을 파울로 처리하고 얼음판에서 제거한다. 파울로 처리된 스톤은 점수 계산 시 반영되지 않는다.

4.2 스윙핑

솔을 이용해 얼음판을 문지름으로써 스톤과 얼음판 사이의 마찰력을 줄여 스톤이 앞으로 더 움직일 수 있도록 하는 스윙핑 동작은 중력 센서의 가속도 센서 기능을 이용

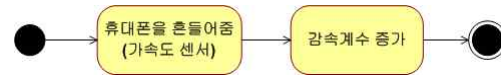
하여 구현하였다. 실제 스위핑 동작을 하듯이 휴대폰을 X축 방향, 즉 가로로 흔들면 가속도 센서를 통해 측정된 X축 방향의 가속도 값을 이용하여 스위핑 기능을 적용한다. 측정된 가속도의 크기에 따라 차별화된 스위핑 효과를 반영하기 위해 가속도 값이 5.0 이상이면 마찰력을 0.02에서 0.008로 줄이고 가속도 값이 2.0에서 4.9사이이면 0.012, 그 외의 경우는 초기 마찰력 값인 0.02를 유지하도록 한다. 스위핑 기능이 적용되면 일시적으로 얼음판 위의 마찰력이 감소되기 때문에 스톤의 이동 속도가 덜 감소한다. 실험에서는 가속도 센서 시뮬레이터를 이용하였다.

휴대폰을 X축 방향으로 흔들어 스위핑 동작을 시도한 경우 가속도 센서를 이용해 가속도 값을 측정하는 모듈과 측정된 가속도 값의 크기에 따라 마찰력 값을 변경하고 그 값으로 스톤의 방향 벡터를 수정하는 모듈의 주요 부분은 다음과 같다.

```
// 스위핑 동작을 위해 가속도 센서의 변화를 감지하고
// 변화된 가속도 값을 저장하는 모듈
public void onSensorChanged(int sensor, float[] values) {
    // 가속도 센서에 변화가 생긴 경우
    if(sensor == SensorManager.SENSOR_ACCELEROMETER)
        // 가속도 센서에 의해 측정된 가속도 값 저장
        for(int i = 0; i < 3; i++)
            mAccelerometerValues[i] = values[i];
}
// 스위핑 동작 후 변경된 마찰력에 따라 스톤의 방향 벡터를
// 변경하는 모듈
private void updateStoneSet() {
    // X축 방향으로 가속도 값이 일정크기 이상 발생한 경우
    // 스위핑 동작으로 처리
    // 더블 스위핑이 발생한 경우 마찰력은 0.008로 설정
    if(Math.abs(mAccelerometerValues[0]) >= 5.0)
        friction = 0.008;
    // 일반 스위핑이 발생한 경우 마찰력은 0.012로 설정
    else if(Math.abs(mAccelerometerValues[0]) >= 2.0)
        friction = 0.012;
    // 스위핑 동작이 실패한 경우 원래 마찰력 값으로 복원
    else friction = 0.02;

    // 변경된 마찰력을 이용하여 스톤의 방향 벡터 수정
    stoneSet[i].setVector(stoneSet[i].getVector().getX()*(1.0-friction),
        stoneSet[i].getVector().getY() *(1.0-friction));
    // 스위핑 동작이 완료되면 원래 마찰력으로 복원
    friction = 0.02;
}
```

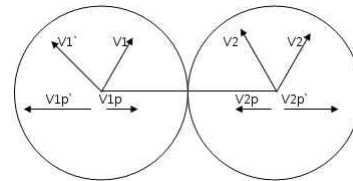
<그림 4>는 스위핑 과정을 설명한다.



<그림 4> 스위핑 과정

4.3 충돌

투구 후 표적판으로 향하는 스톤이 이미 표적판에 안착한 다른 스톤과 충돌하는 경우도 발생한다. 이때 모든 충돌은 완전 탄성 충돌로 가정하여 처리한다. 각각 V1, V2의 초기 방향 벡터를 가지는 두 스톤이 충돌한 후 변경되는 방향 벡터는 <그림 5>와 아래 수식을 이용해 계산한다[6].



<그림 5> 스톤의 충돌 상황

$$V1p' = \frac{(m1 - em2) V1p + (1 + e)m2 V2p}{m1 + m2}$$

$$V2p' = \frac{(m2 - em1) V2p + (1 + e)m1 V1p}{m1 + m2}$$

$$V1' = V1 + (V1p' - V1p)n$$

$$V2' = V2 + (V2p' - V2p)n$$

첫 번째 스톤의 초기 방향 벡터를 V1, 충돌 후 방향 벡터를 V1', 두 번째 스톤의 초기 방향 벡터를 V2, 충돌 후 방향 벡터를 V2', V1을 두 스톤의 중심을 연결한 선에 투영한 벡터를 V1p, V2를 두 스톤의 중심을 연결한 선에 투영한 벡터를 V2p, 두 스톤의 중심을 연결하는 단위 벡터를 n, 탄성계수를 e, 첫 번째 스톤의 질량을 m1,

두 번째 스톤의 질량을 m_2 라 한다. 본 논문에서는 완전 탄성 충돌을 가정하기 때문에 탄성계수는 0이고 두 스톤의 질량은 같다.

스크린에 스톤의 움직임을 표시하는 순간마다 움직이는 스톤과 정지되어 있는 스톤 간의 거리를 계산하여 이 값이 스톤의 지름과 같거나 작아지면 충돌에 의해 변경되는 두 스톤의 진행 벡터를 위 수식에 따라 계산한다.

<그림 6>은 충돌이 발생했을 때의 처리 과정을 설명한다.

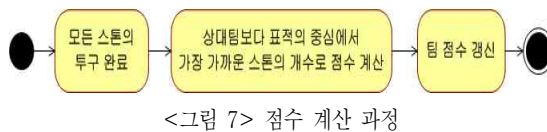


<그림 6> 충돌 처리 과정

4.4 점수 계산

컬링 게임에서는 상대팀 스톤보다 표적판 중심에 가까이 있는 스톤의 개수가 그 팀의 점수가 된다. 따라서 표적판 중심에서부터 각 스톤의 중심까지의 거리를 계산하여 최종 점수를 확정한다. 단, 표적판의 바깥쪽 원 안에 위치한 스톤만이 점수에 반영된다. 3세트가 끝난 후 점수를 계산하여 승리 팀과 패배 팀을 결정한다.

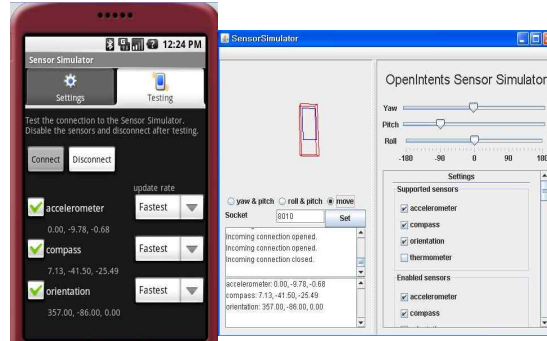
<그림 7>은 점수를 계산하는 과정을 설명한다.



<그림 7> 점수 계산 과정

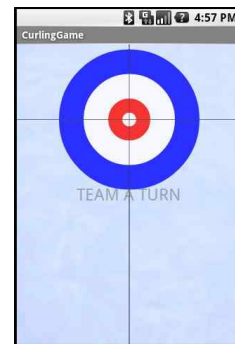
V. 모바일 컬링 게임 구현

<그림 8>은 컬링 게임을 시작하기 전에 가속도 센서 시뮬레이터의 기본적인 사용 환경을 설정하는 화면이다.



<그림 8> 가속도 센서 설정 화면

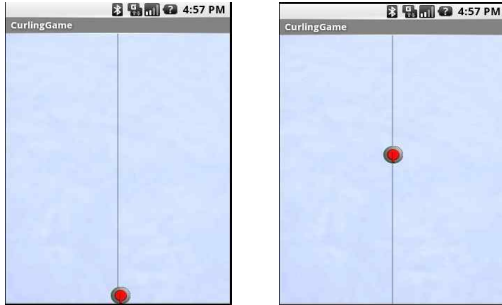
<그림 9>는 컬링 게임을 실행한 후 먼저 공격을 시작하는 A팀의 투구 준비 화면을 보여준다. 화면에는 표적판과 함께 A팀 순서임을 알려주는 메시지가 나타난다.



<그림 9> A팀 투구 준비 화면

<그림 10>은 A팀의 첫 번째 선수가 첫 번째 스톤을 투구한 결과를 보여준다. 투구를 시작하기 위해 처음으로 스크린을 누른 위치의 좌표와 투구를 끝내기 위해 스크린에서 손을 떼 위치의 좌표를 이용해 최종적으로 투구된 스톤의 위치가 결정된다.

<그림 11>은 먼저 공격을 시작한 A팀의 두 번째 투구가 끝난 후 B팀의 두 번째 투구를 기다리는 화면이다. 화면에는 두 개의 A팀 스톤과 한 개의 B팀 스톤이 위치하고 있고 B팀 순서임을 알려주는 메시지가 나타난다.



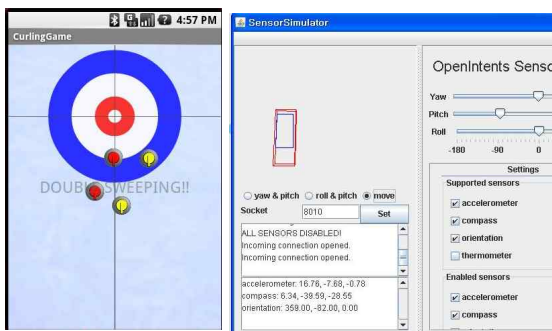
(a) 투구 시작 (b) 투구 완료

<그림 10> 투구 화면



<그림 11> B팀 투구 준비 화면

<그림 12>는 B팀의 두 번째 선수가 투구 후 스위핑 동작을 시도한 결과 화면이다. 가속도 센서 시뮬레이터가 휴대폰의 움직임에 따른 가속도 변화량을 전달받아 스위핑 동작을 처리한다.



<그림 12> B팀의 두 번째 투구 후 스위핑 발생 화면

<그림 13>은 투구한 스톤이 표적판이 위치하고 있는 화면을 벗어나 파울로 처리된 경우를 보여준다. (a)와 같이 화면의 경계선에 스톤이 위치하는 경우도 파울로 처리되며 다음 선수가 투구를 준비할 때 파울로 처리된 스톤은 (b)와 같이 화면에서 제거된다.



(a) 파울 라인에 스톤이 위치한 경우 (b) 파울로 스톤이 제거된 모습

<그림 13> 파울 처리 화면

<그림 14>는 두 팀의 선수들이 모두 한 번씩 투구를 마치고 첫 번째 세트가 끝난 후의 결과 화면이다. 파울 처리된 스톤을 제외하고 5개의 스톤이 표적판 주위에 놓여 있다. 표적판 중심에 가까이 있는 B팀 스톤의 개수보다 A팀 스톤의 개수가 1개 더 많기 때문에 A팀이 1점을 획득했다는 메시지가 화면에 출력된다.



<그림 14> 1세트 종료 후 점수 출력 화면

<그림 15>는 세 번째 세트까지 모두 끝낸 후 최종 결과 화면을 보여준다. 3세트가 진행되는 동안 A팀과 B팀이 획득한 점수는 각각 2점으로 이번 경기가 무승부로 마무리되었다는 메시지가 화면에 출력된다.



<그림 15> 3세트 종료 후 경기 결과 화면

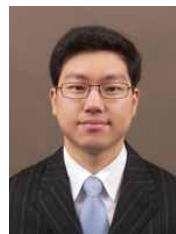
참고문헌

- [1] 김정훈, “구글의 안드로이드와 안드로이드마켓,” 한국콘텐츠학회, 콘텐츠학회논문지, 제7권, 제2호, 2009, pp. 29-36.
- [2] 안드로이드마켓, <http://www.android.com/market/>
- [3] 유희진 · 이면재, “BREW 기반 모바일 게임 개발에 관한 연구,” 한국정보기술학회, 정보기술학회논문지, 제6권, 제3호, 2008, pp. 31-37.
- [4] What is Android?, <http://developer.android.com/guide/basics/what-is-android.html>
- [5] 안드로이드 개발자 페이지, <http://developer.android.com/index.html>
- [6] David Conger, “게임 프로그래머를 위한 물리 모델링,” 사이텍미디어, 2005.

VI. 결론

구글에서 개발한 개방형 모바일 플랫폼인 안드로이드가 탑재된 스마트 폰에 대한 사용자의 관심이 높아지면서 안드로이드 환경에서 구동되는 다양한 애플리케이션에 대한 요구도 증가하고 있다. 특히, 사용자들에게 큰 호응을 받고 있는 게임 애플리케이션의 경우 롤 플레이 게임, 퍼즐 게임, 액션 게임 등의 일반적인 유형을 벗어나는 차별화된 게임 개발이 요구되고 있다. 본 논문에서는 많은 전략적 사고를 필요로 하지만 게임 규칙이 간단해서 누구나 쉽게 즐길 수 있는 컬링 게임을 안드로이드 플랫폼 기반의 애플리케이션으로 개발하였다. 본 논문에서 개발한 안드로이드 기반의 컬링 게임 애플리케이션은 투구, 스위핑, 충돌, 파울 처리 등 실제 컬링 게임에서 발생하는 여러 기능들을 터치스크린과 중력 센서 등의 기술을 이용해 현실감있게 제공하기 때문에 사용자가 실제로 얼음판에서 컬링 게임을 진행하는 것과 같은 만족감과 흥미를 제공한다.

■ 저자소개 ■



조 제 현
Cho, Jae Hyeon

2010년 2월 홍익대학교 컴퓨터공학과(공학사)

관심분야 : 컴퓨터 그래픽스, 게임 프로그래밍
E-mail : cndjrthx@hanmail.net



고 세 환
Ko, Se Hwan

2010년 2월 홍익대학교 컴퓨터공학과(공학사)

관심분야 : 모바일 프로그래밍, 모바일 운영체제
E-mail : york82@hanmail.net



김연희
Kim, Youn Hee

2007년 3월~현재
부천대학 e-비즈니스과
강의전담교수
2006년 8월 홍익대학교 컴퓨터공학과(공학박사)
2002년 2월 홍익대학교 컴퓨터공학과(공학석사)
2000년 2월 홍익대학교 컴퓨터공학과(공학사)
관심분야 : 시맨틱 웹, XML, 분산
데이터베이스, 모바일 데이터베이스
E-mail : yhkim@bc.ac.kr



임해철
Lim, Hae Chull

1981년 3월~현재
홍익대학교 컴퓨터공학과 교수
1988년 8월 서울대학교 컴퓨터공학과(공학박사)
1978년 2월 한국과학기술원
전자계산학과(이학석사)
1976년 2월 서울대학교 계산통계학과(이학사)
관심분야 : 시맨틱 웹, XML, 멀티미디어
데이터베이스
E-mail : lim@cs.hongik.ac.kr

논문접수일 : 2010년 7월 21일
수정일 : 2010년 8월 30일
계재확정일 : 2010년 9월 8일