

# Android를 이용한 스마트폰 게임 어플리케이션

김경하\*, 이애리\*, 최민\*  
 \*충북대학교 정보통신공학부  
 e-mail:rudgk3838@naver.com

## Development of Smartphone Game Application using Android

Kyungha Kim\*, Aeri Lee\*, Min Choi\*

\*Dept of Information and Communication Engineering, Chungbuk National University

### 요 약

스마트폰이 도입된지 불과 몇 년만에 모바일 컴퓨팅은 이제 생활과 밀접한 다양한 콘텐츠를 소비하고 공유할 수 있는 공간이자 도구가 되었다. 이제 우리의 일상은 스마트 폰의 어플리케이션이나 SNS 등을 이용해 보다 합리적인 정보 탐색을 위하여 얻고자 하는 정보를 검색하고 실시간으로 공유한다. 최근 선호하는 스마트폰 어플리케이션은 오락, 유틸리티, 소셜 네트워크의 비중이 크다. 본 논문에서는 이러한 스마트 폰의 흐름에 발 맞춰, Android기반 게임 어플리케이션을 구현하고자 한다. 본 어플리케이션은 컴퓨터에서만 즐길 수 있던 게임에서 벗어나 보다 접근성이 편리한 스마트폰에서 실행할 수 있을 뿐 아니라 커뮤니케이션 인프라(별점 주기, 랭킹)와 item shop 기능을 제공한다. 본 연구에서는 더블버퍼링, 각종 센서, Surface View 등을 활용하여 스마트폰 게임 어플리케이션을 제작한다.

### 1. 서론

본 연구에서는 가속도 센서를 이용한 안드로이드 게임 어플리케이션을 개발, 연구하는데 있다. 스마트폰이 대중화 되면서 사람들의 삶의 방향은 변화하고 있다. 소셜네트워크 카카오톡, 라인, 마이피플 같은 어플리케이션이 스마트폰의 주 사용목적으로 작용하므로써 소셜네트워크와 다른 목적의 어플리케이션을 접목하는 것들이 다양하게 생기고 있다. 특히, 최근에는 게임을 접목하여 사람들에게 새로운 자극과 흥미를 주고 있다. 소셜네트워크로 사람들과 연락을 주고 받는 것 뿐아니라 게임을 접목하여 게임에 필요한 정보를 서로에게 메시지를 남김으로서 의사소통을 할 수 있는 새로운 방법으로 떠오르고 있다. 따라서, 본 연구에서는 스마트폰의 가속도 센서를 이용한 안드로이드 앱을 개발하고 기존에 존재하는 게임들과는 다르게 새로운 방향으로 게임을 제작하고자 한다.

### 2. 개발 주안점

#### 1) 더블 버퍼링

보통 Android Application은 GDI Thread를 통해 화면에 그려지게 된다. 하지만 빠른 속도와 안정적인 게임 화면 제어를 위하여, 'SurfaceView' Interface를 상속 받아 'SurfaceView.Holder' Callback함수로 작업을 진행한다. 또한 Thread를 상속받은 inner Class를 선언하여, 화면에 그려질 백그라운드, 캐릭터, 몹, 아이템 이미지 등을 고속의 비동기 작업을 통하여 제어된다.

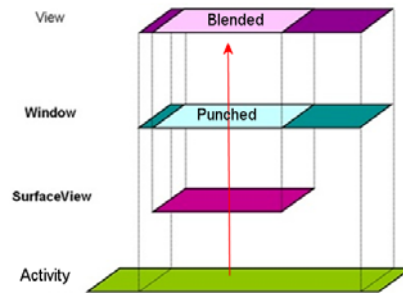


그림 1 surface view 요소

SurfaceView Class는 기본적으로 SurfaceView(Context context) 생성자와, void Draw(Canvas canvas), SurfaceHolder.getHolder(), void surfaceChanged(SurfaceHolder holder, int format, int width, int height), void surfaceCreated(SurfaceHolder holder), void surfaceDestroyed(SurfaceHolder holder)를 기본 제공하여 override하여 사용한다.[2]

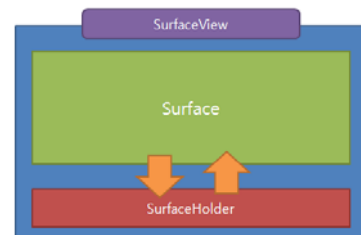


그림 2 surface view

SurfaceView의 구조이다. Android는 작업 로딩이 5초 이

상이 넘어가면 ANR을 발생시켜 강제 종료시킨다. 특히 이미지를 불러오는 과정에서 많은 메모리와 시간을 소비하기 때문에 ANR을 발생 시키는데, 이를 방지하기 위해 SurfaceView가 사용된다.

그림 4.1을 보면 SurfaceView는 SurfaceView.Holder 라는 Callback함수로 Surface 버퍼에 그림을 그리게 되고, 그것이 SurfaceView에 반영이 되어 View에 표시되는 방식이다. 처리속도가 빠른 메모리에서 이미지를 처리하여 메모리에서 View로 바로 전송이 되는 것이다.

2) 안드로이드 가속도 센서를 이용한 제어

Android Device에 내장되어 있는 가속도 센서를 이용하여 사용자가 X, Y, Z축에 따른 Device의 움직임을 감지한다. 가속도 센서에서 넘어오는 감도 값을 게임 내 캐릭터를 움직이게 해준다.

3) 충돌처리 및 각종 센서를 활용한 캐릭터 제어

게임 충돌 처리 기법중 바운딩 박스 충돌 체크를 이용한다. 게임 캐릭터와 몹이나 아이템은 하나의 박스로 생각할 수 있기 때문에 바운딩 박스 충돌 체크를 이용하게 된다. Android Device에 내장되어 있는 가속도 센서를 이용하여 사용자가 X,Y,Z축에 따른 Device의 움직임을 감지한다. 가속도 센서에서 넘어오는 감도 값을 게임내 캐릭터를 움직이게 해준다. .

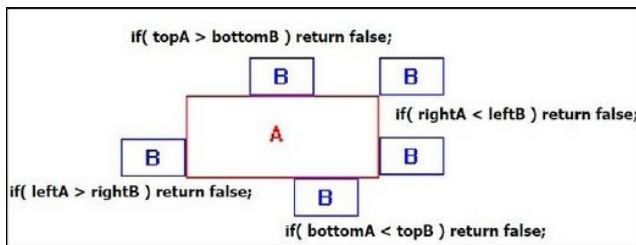


그림 3 충돌처리 알고리즘

A객체와 B객체가 충돌 할 때 사방면을 체크하게 된다. 한 객체를 기준을 잡고 왼쪽 오른쪽 위 아래를 if()문을 통해 체크를 하게되며, 각 박스의 중점을 이었을 때 일정 크기를 이용하여 체크한다.

① 몹 : 몹을 주인공 캐릭터의 움직임에 반응하여 움직이게 구성하였다. 몹1은 주인공 캐릭터를 피하면서 움직이게 하여 좌우로 피하게 되고, 몹2는 주인공 캐릭터를 향하여 움직임으로써 피해를 입히게 하는 구성으로 되었다.

주인공 캐릭터의 x좌표값을 계속 알아내어 x좌표를 기준으로 몹의 x좌표와 같으면 좌또는 우로 움직이게 하였고, x좌표보다 작으면 좌로, 크면 우로도 하게끔 하였다. 몹2는 반대로 하여 적용하였다. 그리하면 주인공 캐릭터를 따라 움직이게 된다.

② 아이템 : 첫 번째 아이템은 대각선으로 움직여서 벽에

충돌하면 반대방향 대각선으로 움직이게 된다. 두 번째 아이템은 일직선으로 움직이게 하였다.

아이템을 주인공 캐릭터가 캐치하게 되면, flag변수를 두어 y좌표를 -y좌표로 바꾸어 위로 올라가게 하게 구성하였다.

Android Application에서 배경음악과 효과음은 일반적으로 MediaPlayer()메서드를 이용한다. 하지만, 배경음악처럼 긴 시간동안 load되는 경우는 MediaPlayer()메서드가 적합하지만, 효과음은 짧게 플레이 되는 부분이라 메모리 관리를 위하여, SoundPool()메서드를 이용한다.

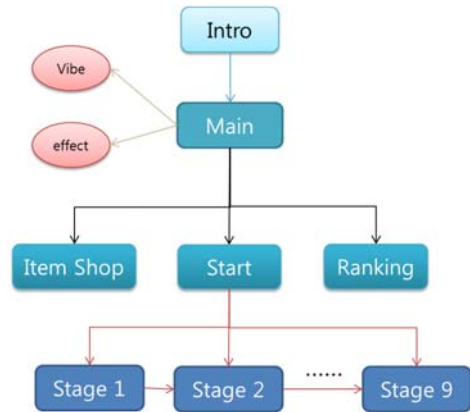


그림 4 Activity 흐름도

반 사용자에게 보다 편리한 사용자 인터페이스 환경을 제공하기 위해서는 현재의 윈도우즈의 기반 사용자 인터페이스의 차원을 넘어서 사용자의 작업을 대행해 줄 수 있는 에이전트 시스템이 제공되어야 한다. 또한 에이전트 시스템서비스 확장과 사용보급을 위하여 응용을 위한 미들웨어 플랫폼에 대한 연구개발이 이루어져야 한다.

3. 결론 및 향후연구

스마트폰이 일반화 되면서 그에 따라 유행하고 있는 부분을 연구해 보았다. 그것은 소셜 네트워크와 접목하여 인기를 끌고 있는 게임관련 개발이다. 현재 존재하고 있는 것들과는 사뭇 다르게 연구를 해보았고 그 결과는 새로웠고 나쁘지 않았다. 개발을 통해서 전체적인 안드로이드 사용법을 숙지 할 수 있었고, 다른 쪽으로도 앱을 개발할 수 있는 자신감도 얻었다. 아직은 수정, 보완이 필요한 상태이고 이를 통해 더 세심하고 고급스럽게 하려는 노력이 필요하다. 향후 어플리케이션과 서버를 연동 시켜야 하는 과제가 남아있다. 서버까지 완성이 된다면 더 게임의 질이 높아질 것이라는 기대감을 가지고 있다.

참고문헌

[1] Roger S. Pressman "Software Engineering A Practliners' Approach" 3rd Ed. McGraw Hill