

화면 중심 안드로이드 어플리케이션 저작도구의 설계

김규완*, 김우빈, 이은정
*경기대학교 컴퓨터과학과

e-mail: kkw5240@gmail.com, woobinkim508@naver.com,
ejlee@kyonggi.ac.kr

Design of an authoring tool for scene-oriented Android application

Kyu-wan Kim*, Woo-bin Kim, Eunjung Lee
*Dept of Computer Science, Kyonggi University

요 약

스마트폰이 사회적으로 보편화 되면서, 어플리케이션 사용량이 증가하고 있다. 누구나 사용하는 어플리케이션이지만 어플리케이션을 만드는 것은 상대적으로 소수인 개발자들만이 가능한 것이 현실이다. 따라서 본 논문에서는 누구나 간단하게 위지윅(WYSIWYG) 기반의 인터페이스를 이용하여, 화면과 터치 이벤트로 구성되며 사용자와 상호작용이 풍부한 화면 중심의 안드로이드 어플리케이션을 제작 할 수 있는 저작도구를 설계하였다.

1. 서론

스마트폰이 사회적으로 보편화 되면서 사용하는 어플리케이션의 개수와 사용 시간 또한 증가하고 있다. 어플리케이션에 대한 관심이 높아짐에 따라 자신이 직접 어플리케이션을 제작해 보고 싶어 하는 사람도 많이 생겼지만 실제로 별도의 교육을 받은 적 없는 일반인이 어플리케이션을 제작하기에는 어려움이 많은 것이 현실이다. 이에 따라서 다양한 종류의 저작도구들이 나왔지만, 이러한 저작도구들은 여전히 앱 구성 측면에서 정해진 모양과 방식대로만 제작 할 수 있어 사용자가 만족할만한 수준의 어플리케이션을 제작하는데 문제가 있었다.

본 논문에서는 위지윅(WYSIWYG) 기반의 인터페이스를 제공하여 쉽게 이미지와 버튼으로 화면을 구성하고, 화면 간 자유로운 연결을 통하여 풍부한 상호작용과 시각적 효과를 가지는 화면 중심의 어플리케이션을 제작 할 수 있는 저작도구를 제안한다.

저작도구에서 제작한 결과물은 xml 파일로 저장되며 실행 라이브러리와 함께 패키징되어 간단한 구조의 어플리케이션을 생성한다.

제안된 방식은 코딩 없이 쉽게 이미지와 터치 이벤트로 단순한 구조의 어플리케이션을 제작할 수 있으며, 다양한 화면 간 이동방법(Navigation)을 통하여 풍부한 상호작용과 시각적 효과를 얻을 수 있다.

본 연구는 미래창조과학부 및 정보통신산업진흥원의 산학협력 특성화 지원사업의 연구결과로 수행되었음
(NIPA-2014-H0806-14-1004)

2. 기존의 저작도구들

현재 사용 가능한 공개된 저작도구 중 코드가 들어가지 않는 방식의 저작도구는 대부분 전자책을 목표로 하는 것이 많다. Namo Pubtree[1], studio4ux[2], ViewPorter[3] 등은 모두 전자책을 위한 저작도구로서 화면 간 이동이 제한적이고, 정적인 화면만을 제공하는 것이 대부분이다.

일반 어플리케이션 저작도구인 '앱쿠커[4]'는 쉬운 UI와 편리성으로 초보자들도 쉽게 어플리케이션을 만들 수 있도록 구현되어 있지만 탭과 리스트뷰의 컨트롤로 구성된 서식만을 제공하므로 다양한 구조의 어플리케이션을 제작하는데 한계가 있다.

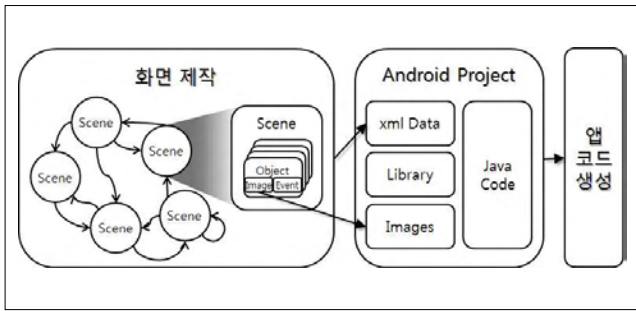
비즈니스 로직 처리 저작도구인 m-Bizmaker[5]는 기본적인 버튼기능부터 DB까지 다양한 기능들이 들어있어 코드의 개발 없이도 비즈니스 로직 처리가 가능한 어플리케이션을 만들기 위하여 많이 사용하지만 다양한 기능을 제공해야 하기 때문에 m-Bizmaker로 제작된 어플리케이션을 스마트 디바이스에 설치할 때, 별도의 엔진 역할을 하는 m-BizEngine을 따로 설치해서 사용해야 하므로 무겁고 속도가 느리다.

본 논문에서는 기존 도구들의 문제점을 해결하면서 동적인 화면 구성이 가능하고, 화면 간 이동에 제한이 없어 사용자와의 상호작용이 활발한 어플리케이션을 제작할 수 있는 저작도구를 설계 및 구현하고자 한다.

3. 설계

본 저작도구의 전체 구조는 <그림 1>과 같으며 기능은

크게 어플리케이션 저작, 데이터 생성, 패키징 및 APK 파일 생성으로 나뉜다.



[그림 1. 저작도구의 전체 구조]

어플리케이션은 여러 개의 Scene으로 구성된다. Scene은 Object들로 구성된 어플리케이션의 한 화면을 의미하며, 자체적인 이벤트를 가질 수 있다. 모든 내비게이션은 이벤트를 통하여 다음 Scene을 지정함으로써 설정한다. Object는 화면을 구성하는 가장 작은 단위이며, 이미지와 이벤트로 구성된다.

이벤트는 Object와 Scene에서 적용 할 수 있으며, Object에서는 터치 이벤트를 적용 할 수 있고, Scene에서는 Scene 내부의 모든 애니메이션이 종료 된 경우에 이벤트를 적용 할 수 있다.

3.1. 어플리케이션 저작

어플리케이션 저작 단계에서는 사용자가 에디터를 통해 이미지를 등록하여 화면을 구성하고, 등록된 이미지에 이벤트를 추가하는 방식으로 어플리케이션을 설계한다. 화면에 추가된 이미지는 위치, 크기, 회전각도, 애니메이션 등을 설정 하여 화면을 꾸밀 수 있다.

이벤트가 추가된 이미지 또는 화면의 경우, 이벤트를 설정하면 이벤트가 발생했을 때 원하는 화면으로의 이동(Navigation)을 자유롭게 지정할 수 있다. 이를 통하여 어플리케이션의 다양한 흐름을 만들게 된다.

이렇게 설계된 화면은 실제 어플리케이션에 적용되는 화면과 동일하게 보이며, 에디터의 시뮬레이션 기능을 통하여 실제로 테스트해 볼 수도 있다.

3.2. 데이터 생성

데이터 생성 단계에서는 저작 단계에서 설계된 어플리케이션 화면 제작 결과를 xml형식으로 저장한다. 아래 <표 1>은 본 저작도구에서 생성되는 화면 이동 정보(Scene Navigation Data)의 중요 값이다. 이렇게 저장된 화면 정보들은 제작된 Android 어플리케이션이 실행될 때 이용된다.

구성	종류	정의
Object	ID	Object의 ID
	ImageTag	이미지의 ID
	Position	이미지의 가운데 위치좌표
	Size	이미지의 크기
	Animation	Object의 애니메이션 정보
	Event	이벤트의 종류 및 연결된 화면의 ID
Scene	ID	Scene의 ID
	BG_Color	Scene의 배경 색
	BG_IMG	Scene의 배경 이미지
	ObjList	Scene이 가진 Object들
	Event	이벤트의 종류 및 연결된 화면의 ID

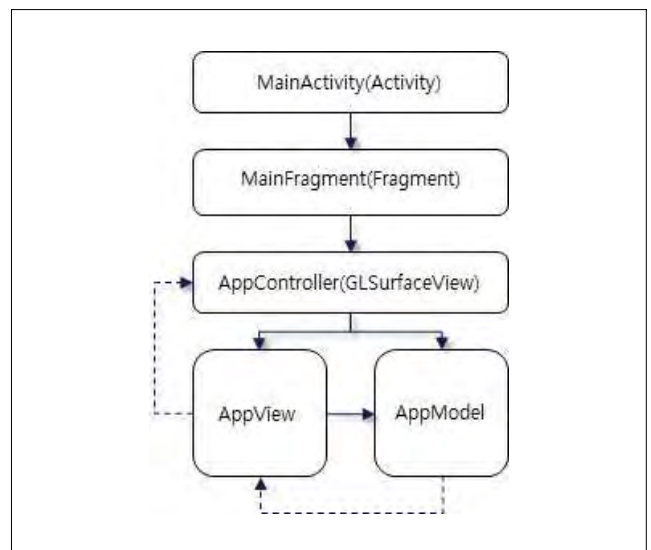
[표 1. SceneNavigation Data의 중요 값]

3.3. 실행 환경 라이브러리 및 클래스 설계

실행 환경 라이브러리에서는 SNData(SceneNavigation Data)에 저장된 정보들을 읽고, 이를 이용해 안드로이드 환경에서 OpenGL ES로 화면을 구성 하는 함수들을 제공한다.

실행 환경 라이브러리는 크게 AppModel, AppView, AppController 3개의 클래스로 구성되어 있다. AppModel은 SNData에 저장된 정보들을 읽어 저장하고, 그 정보들을 바탕으로 화면 구성에 필요한 정보들을 제공한다. AppView는 OpenGL ES의 Renderer 클래스를 상속받아 어플리케이션의 화면을 OpenGL ES의 함수를 이용하여 그리는 역할을 한다. 마지막으로 AppController는 GLSurfaceView를 상속받아 App의 이벤트를 처리하고 AppModel에서 필요한 정보를 받아 AppView에 전달하여 화면을 그리게 한다.

그 외에 MainActivity가 있으며 MainActivity에서 AppController 클래스를 콘텐츠뷰로 설정하여 라이브러리를 이용하도록 한다. 이상의 클래스 구조가 [그림]에 나타나 있다.



[그림 2. 저작된 Application의 구조도]

```

class AppController extends GLSurfaceView
void function init
    DataManager = appModel.read( XML File );
    currentScene = DataManager.sceneList.get(0);
    AppView.show( currentScene );
end function

boolean function onTouchEvent
    Position = Event.Position;
    for( Object obj : currentScene.objList )
        if( Position in the obj.position )
            if( obj.hasEvent )
                currentScene = DataManager.get(object.nextScene);
                AppView.show( currentScene );
            end function
    end function

```

[그림 3.AppController Class]

```

class AppView( Scene scene ) extends Renderer
setting OpenGL ES Enviroments

void function show( Scene scene )
    for( Object obj : scene.objList )
        image = getTexture(obj.imageTag);
        draw image with OpenGL ES
    end function

```

[그림 4.AppView Class]

그 결과 xml 형식의 파일을 읽어서 버튼 이벤트 처리를 하는 단순한 구조를 가진 어플리케이션을 만들 수 있도록 하였다. 별도의 코딩 없이 제작된 단순한 어플리케이션이지만 다양한 애니메이션과 자유로운 화면 구성을 통해 복잡한 구조로 화면 이동이 가능하도록 하여 풍부한 상호작용과 높은 시각효과를 줄 수 있을 것으로 기대된다. 또한 그래픽 처리 시, 발생 할 수 있는 속도 저하를 OpenGL ES[6]의 파이프라인 연산을 통하여 해결한다.

4. 제안된 방법의 앱 적용사례

제안된 방법은 화면과 객체, 그리고 이벤트에 따른 화면 이동만으로 구성되어 누구나 이해할 수 있고 쉽게 설계할 수 있는 앱 구조를 가진다. 또한 제안된 저작도와 실행 환경을 이용하면 코드의 개발 없이 쉽게 앱을 생성할 수 있다. 더구나 이러한 단순한 구조에도 불구하고 많은 수의 화면을 가지고 이동 관계가 복잡하게 구성되어 있는 앱을 작성할 수 있으며, OpenGL ES의 기능을 이용하여 풍부한 상호작용과 애니메이션 효과를 넣을 수 있다.



[그림 5. 방 탈출 게임]

[그림 5]는 <http://www.3wish.com/game/game.htm> 사이트의 플래시게임이다. 본 논문에서 제안한 방법을 이용하여 게임을 설계한다면 저작도구에서 쉽게 모바일 어플리케이션으로 제작 가능할 것이다.

사용자는 화면과 객체의 이미지를 작성한 후 제안된 저작도구를 이용하여 객체의 상호작용과 화면 이동을 자유롭게 설계할 수 있으며, 방문이나 서랍의 열림 등의 애니메이션 효과를 추가 할 수 있다.

5. 결론

본 논문에서는 이미지와 애니메이션, 내비게이션으로 구성되는 모바일 어플리케이션 구조를 제안하고 그러한 앱 제작을 위한 코드 없는 저작도구의 제안하였다.

에디터에서는 OpenGL기반으로 화면을 구성하여 애니메이션 효과를 줄 수 있도록 하였다. OpenGL을 이용하여 OpenGL의 파이프라인 연산을 이용해 빠른 속도로 효과를 적용 할 수 있도록 하였고, 차후에 모바일 어플리케이션으로 적용이 용이하도록 하였다. 또한 화면 간 내비게이션 기능을 자유롭게 할 수 있도록 하여 보다 풍부한 상호작용이 가능하도록 하였다. 그리고 설계된 어플리케이션을 미리 볼 수 있는 시뮬레이션 기능이 있어, 실제로 모바일에서 어떻게 실행되는지 설계 도중 확인 해 볼 수 있도록 하였다.

제안된 방법을 이용하면 누구나 쉽게 '방 탈출 게임', '심리테스트' 등의 어플리케이션을 만들 수 있을 것으로 기대된다. 현재 시스템 설계와 에디터의 구현이 진행 중이며, 안드로이드 런타임 클래스를 구현 할 계획이다.

6. 참고 자료

[1] Namu. 2014. "나모 펍트리 에디터." 2014.09.14. <http://new.namo.co.kr/site/namo/menu/5044.do>

- [2] Net&tv “Studio4ux is.”
http://studio4ux.co.kr/studio4ux_kr.php
- [3] OrangeDigit “ViewPorter-Thunder.”
<http://viewporter.com/ko/viewporter-thunder/>
- [4] CAMDENSOFTE “Appcookr.”
<http://www.appcookr.co.kr/Features/Index>
- [5] WonderfulSoft “m-Bizmaker”
<http://www.appcookr.co.kr/Features/Index>
- [6] Khronos “OpenGL ES”
<http://www.khronos.org/opengles/>
- [7] 권순옥
“스마트 디바이스를 위한 교육용 콘텐츠
저작 도구 설계 및 구현.”
한국콘텐츠학회논문지 ‘13 Vol. 13 No. 12
- [8] 장연현
“저작도구를 활용한 비전공자의 스마트폰
어플리케이션 개발 효율성에 대한 실험적 고찰.”
한국컴퓨터정보학회 하계학술대회 논문집 제19권 제2호
- [9] 백낙훈
“데스크탑 상에서의 OpenGL ES 2.0 에뮬레이션”
정보처리학회논문지. 컴퓨터 및 통신시스템.
Vol. 3 No. 4 (2014) pg. 125, 4 p.