

# 모바일 앱을 위한 동적 텍스트 편집 저작도구 설계

강용희\* 이은정  
경기대학교 컴퓨터과학과  
e-mail : AL123SK@naver.com

## Design of Dynamic Text Authoring Tool for Mobile Applications

Yong-Hee Kang\* Eun-Jung Lee  
Dept of Computer Science, Kyonggi University

### 요 약

모바일 디바이스 보급이 확산되고 다양한 앱을 보다 쉽게 만들 수 있게 도와주는 저작도구들이 발표되었으나 모바일 앱을 위한 텍스트의 디자인과 효과를 편집할 수 있는 저작도구는 부족하다. 본 논문에서는 상호작용과 시각적 효과를 높일 수 있는 텍스트 편집 기능을 가지면서 설계 결과를 앱에 통합하기 위한 저작도구를 제안한다. 이 저작도구를 통하여 사용자는 보다 편리하게 텍스트를 편집하고 실행 화면을 미리 확인할 수 있다. 설계된 결과는 xml 데이터로 저장되고 이를 모바일 디바이스에서 실행하기 위하여 실행환경의 텍스트 처리 라이브러리 클래스를 제공하고 이것을 앱의 개발자 코드에 통합하는 방법을 제시한다.

## 1. 서론

### 1.1 동기

오늘날 사회는 모바일 디바이스의 보급률이 급속히 증가되고 있으며, 이를 기반으로 하여 모바일 디바이스에 대한 다양한 앱이 출시되고 있다. 또한 시간이 지날수록 청소년들도 간단한 앱을 만들어 사용할 수 있도록 하기 위한 저작도구들이 증가하고 있다.[1][2][3]

이 저작도구들은 플랫폼에 따라 다양하게 존재하며 그 기능들 또한 간단한 이미지 편집부터 논리 식을 사용하게 하거나 멀티미디어 기능을 지원하기도 한다. 본 논문에서는 모바일 디바이스 플랫폼 환경에서 애니메이션 텍스트를 사용자가 위지윅(WYSIWYG)방식으로 간단하게 제작할 수 있도록 도움을 주는 모바일 디바이스용 애니메이션 텍스트 저작도구에 대한 설계를 한다.

### 1.2 목적

본 논문은 모바일 디바이스에서 사용되는 텍스트에 대하여 에디터를 사용함으로써 개발자가 아닌 이들도 텍스트를 편집할 수 있도록 한다.

에디터에서는 텍스트에 시각적 디자인뿐만 아니라 상하 좌우로 움직이는 애니메이션을 편집함으로써 모바일 디바이스에서 텍스트에 시각적인 효과를 부여할 수 있다.

또한 이렇게 만들어진 데이터를 개발자가 플랫폼에서 코드화 시키는 번거로움을 줄이는 방법을 제시한다.

## 2. 관련 연구

본 논문에서 지향하는 바와 관련된 저작도구로는 UCNovel[4]과 studio4ux[5]가 있다. 이 도구들은 직관적인 위지윅(WYSIWYG) 방식의 사용자 인터페이스 화면을 제공하며 UCNovel은 웹 콘텐츠용, studio4ux는 모바일 앱용 저작도구이다.

먼저 UCNovel은 웹 소설 저작도구로서, 해당 사이트 내에서만 배포 및 사용이 가능하다. 이 저작도구의 특징으로는 배경화면의 자유로운 선택과 여러 가지 이미지의 추가 및 텍스트의 조정이 가능하다는 것이다. 사용자들은 해당 UCNovel 사이트의 저작도구를 다운로드 한 후 간단한 조작만으로 웹 소설을 만들 수 있으며, 코드를 전혀 사용하지 않아도 된다는 장점이 있다. 하지만 이 저작도구는 해당 사이트 내에서만 사용이 가능하다는 것과 모바일 기기로 적용되지 않는다는 한계성이 존재한다.

다른 저작도구로는 studio4ux가 있다. 이 저작도구는 OpenGL ES 2.0기반의 디지털 콘텐츠 저작도구이며, 앱북과 이북 제작에 적합하다. 다양한 이미지, 동영상, 텍스트에 대하여 제작이 가능하며 저작도구에서 데이터 파일을 생성한 후 엔진을 통하여 앱이 만들어지지만, 만들어진 텍스트의 사용이 앱북과 잡지 형식에만 제한되는 한계성을 보인다.

### 3. 설계

본 논문에서는 앞서 설명한 기존의 저작도구들의 한계를 해결하기 위해서 새로운 PC기반의 텍스트 저작도구와 모바일 플랫폼 설계를 제안한다. 제안하는 저작도구는 모바일 플랫폼에 적용하기 전 텍스트 편집이 가능하며 간단한 애니메이션을 적용할 수 있다..

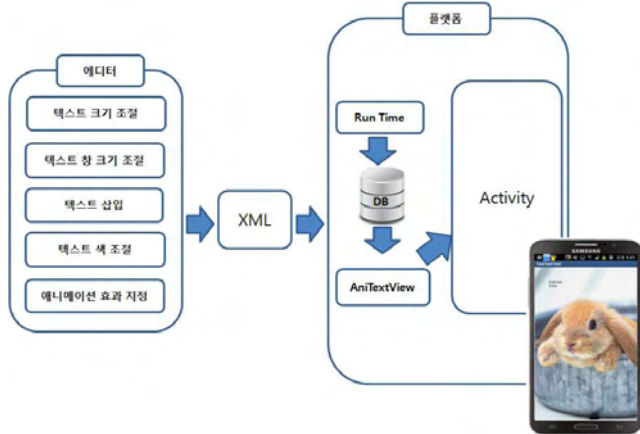


그림 1. 텍스트 저작도구 구조

#### 3.1 에디터

본 논문에서 제안하는 텍스트 저작도구의 에디터에서는 사용자가 모바일 디바이스 상에서 보이는 화면과 동일한 화면을 제공하며, 모바일 플랫폼과 같은 상태에서 텍스트 편집을 가능하도록 해준다. 텍스트 삽입과 크기 제어, 색과 폰트 제어 및 간단한 애니메이션이 가능하도록 만든다. 텍스트 객체 편집 및 앱 개발 과정은 [그림 1]과 같다. 사용자는 텍스트를 생성하고 해당 텍스트에 다양한 편집(크기 조절, 색 조절, 애니메이션 효과 지정)을 할 수 있다. 에디터에서 수행한 편집 정보들은 xml[6]로 변환되어 저장된다.

애니메이션 기능은 미리 지정된 효과 중에서 사용자가 선택하게 되는데 나타나기와 사라지기 등이 있으며 각 효과에 대해 사용자는 상하좌우의 옵션을 지정할 수 있다. 해당 효과에 대한 설정은 파워포인트[7] 편집기와 유사한 대화상자로 구현할 수 있다.

#### 3.2 XML 데이터 저장

편집기에서 설계한 결과는 모바일 플랫폼에서 사용 가능한 xml의 형태로 저장할 수 있다. xml에는 에디터에서 편집한 정보(텍스트뷰 크기, 텍스트 크기, 색, 애니메이션, 삽입한 텍스트)들이 기록된다. 이 정보들은 모바일 플랫폼에서 읽어 화면에 렌더링하기 위하여 사용된다.

애니메이션 정보는 사용자가 선택한 효과의 종류와 그에 대한 옵션으로 저장된다. 예를 들면 텍스트가 오른쪽으로 흐르는 효과의 경우 속도가 0~3 사이로 설정될 수 있으며 그 경우 해당 데이터가 저장되는 방식을 xml 형태로 표현하면 다음과 같다.

```
<SQLdata>
<SetText>
  <firstname>FirstText</firstname>
  <secondname>Title</secondname>
  <size>30</size>
  <width>139</width>
  <height>50</height>
  <color>GREEN</color>
  <font>굴림</font>
  <animation>None</animation>
  <course>None</course>
  <speed>0</speed>
  <text>정발산</text>
</SetText>
<SetText>
  .....
</SetText>
</SQLdata>
```

#### 3.3 모바일 플랫폼

모바일 플랫폼에서는 xml에 저장된 텍스트 정보 (텍스트뷰 크기, 텍스트 크기, 색, 애니메이션, 삽입한 텍스트)를 가져와야 한다.

xml에 저장된 정보를 먼저 데이터베이스에 구축하기 위해서 xml의 경로를 탐색한 후 SQLite에 xml의 정보를 삽입한다. 개발자는 구축된 데이터베이스에서 키값으로 검색하여 해당되는 xml의 문서를 처음부터 끝까지 오면서 xml에 저장된 텍스트를 배열로 읽어 차례로 데이터베이스에 입력하는 식으로 구현한다.

```
SQLiteDataBase db; //DB 선언
XmlResourceParser xml = res.getText(Xml 경로);
while(eventType != XmlPullParser.END_DOCUMENT)
{
  // 프로젝트 DB 불러오기
  DB dictionary = new DB(this);
  //필요한 데이터 입력
  db = dictionary.getWritableDatabase();
  ContentValues values = new ContentValues();
  values.put( firstname, xml.getText() );
  values.put( seceondname, xml.getText() );
  .....
  //레코드 추가
  db.insert( 테이블 이름, null, values );
  db.close(); //db 종료
}
```

텍스트 객체를 화면에 배치할 때 안드로이드 환경의 표준 라이브러리 중 TextView[8] 클래스의 속성 지정 기능을 사용할 수 있다. 그러나 설계된 결과를 화면에 렌더링하는 일을 앱의 다른 코드와 쉽게 통합하기 위하여 본 연구에서는 실행 환경에 맞는 라이브러리 클래스 AniTextView를 개발한다. 이 클래스는 1) 데이터베이스의 해당 객체 값을 읽어오고, 2) 화면에 속성에 따라 텍스트를 표시하고, 3) 애니메이션 효과 등 TextView 클래스가 제공하지 못하는 기능을 지원해야 한다.

안드로이드 앱의 개발자 코드에서는 제공된 라이브러리 클래스 AniTextView를 이용하여 다음과 같은 과정을 거쳐 설계된 텍스트 객체를 액티비티에 추가하게 된다.

```
//firstname,secondname을 기본 키로 하여 AniTextView
//에서 SQLite를 열어 텍스트뷰 생성
AniTextView aniText =
    new AniTextView(this,"FirstText","Title");
aniText.setLayoutParams(parm); //크기 조정
layout.addView(aniText); //현재 레이어에 텍스트뷰 삽입
aniText.setX(x); //좌측 상단(0,0)을 기준으로 x위치
aniText.setY(y); //좌측 상단(0,0)을 기준으로 y위치
```

4. 구현

이 절에서는 제안하는 저작도구를 구현하기 위한 에디터와 모바일 디바이스의 실행환경에 대해 기술한다.

제안하는 애니메이션 텍스트 저작도구의 에디터는 Windows8 시스템의 환경에서 Visual Studio 2013(C#)과 Eclipse Juno(Android)를 이용하여 구현중이다.

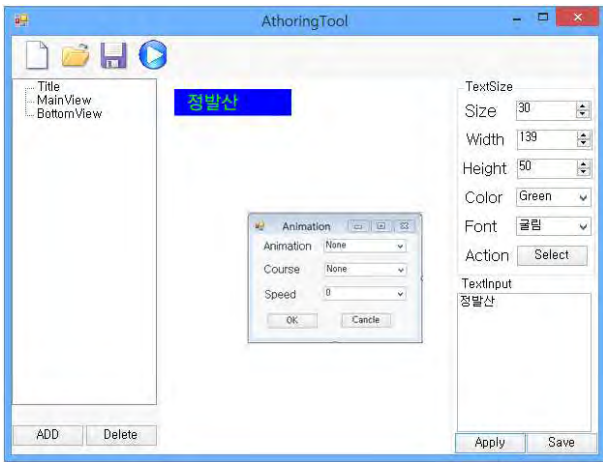


그림 2. 에디터 화면

현재 C#을 이용하여 구현된 에디터에서는 텍스트 삽입과 크기 조정, 색 조정이 가능하다.

[그림 2]의 에디터에서 보이는 것과 같이 텍스트의 크기와 색 폰트의 조정이 가능하고 Action열의 Select버튼을 클릭하게 되면 Animation이라는 창이 하나가 뜨게 된다. 여기서 사용하려는 Animation효과와 그 효과에 해당하는 옵션을 고르고 마지막으로 속도를 조절하게 되면 텍스트 편집이 완료된다.

id	firstname	secondname	size	width	height	color	font	animation	course	speed	text
1	FitsText	Title	30	139	50	GREEN	굴림	None	None	0	정발산
2	FitsText	MainView	15	350	900	BLACK	굴림	None	None	0	일산 신시가지의 중앙공-
3	FitsText	BottomView	30	720	90	GREEN	굴림	Appear	Left	2	호수공원에서 정발산까지
4	SecondText	Title	40	240	40	BLUE	굴림	Appear	Right	2	한라산
5	SecondText	LeftText	30	360	300	BLACK	굴림	Disappear	Left	1	한라산 등산 코스 안내
6	SecondText	RightText	30	360	300	RED	굴림	Disappear	Right	3	제주 관광 명소

그림 3. 안드로이드에 삽입되는 데이터베이스

[그림 3]에서는 에디터에서 생성한 xml를 가져와 SQLite로 변환시킨 결과물이다. 에디터에서 생성한 firstname, secondname와, 텍스트 크기, 텍스트뷰 크기, 색, 폰트 순으로 들어 있으며 그 뒤를 이어 애니메이션 효과, 옵션, 속도 그리고 텍스트 정보로 구성된다.

안드로이드를 이용하여 구현된 모바일 플랫폼 부분에서는 AniTextView 클래스에서 텍스트뷰 생성 및 위치, 크기

조정, 텍스트 삽입, 크기, 색 조정이 구현중이다. SQLite[9]에서 생성한 데이터베이스에서 firstname, secondname으로 아래 코드와 같다. 텍스트뷰의 크기와 텍스트의 크기 색 폰트 및 텍스트 값을 넣어서 텍스트뷰를 생성 한다.

```
Class AniTextView(string firstname, string secondname)
{
    //firstname과 secondname을 조건으로 하여 DB검색
    Cursor cursor = db.query( tablename, columns[], select, null., null, null, null);
    //텍스트 크기조정
    LayoutParms parm = new LayoutParms( width, height );
    //텍스트 색, 크기, 폰트 조정
    setTextColor( color );
    setTextSize( size );
    setFont( font );
    setText( text ); //텍스트 삽입
    //애니메이션
    new Animation(this. speed, animation, course);
}
```

그 후 애니메이션 효과와 옵션에 대하여 Thread를 이용하여 애니메이션 텍스트에 대하여 동적인 움직임을 부여할 수 있다. 이때 Thread의 Sleep시간은 데이터베이스의 speed에 의하여 결정되고, speed가 0이면 애니메이션 효과가 사용되지 않으며 3으로 갈수록 애니메이션 효과가 빨라진다. 즉 효과와 옵션에 대하여 출력 속도는 speed에 의하여 결정되며 speed가 0일 경우는 아무런 효과도 발생하지 않게 되고, speed가 0이 아닐 경우 Thread를 이용하여 출력시간을 조정함으로써 텍스트가 움직이는 효과를 줄 수 있다.

[그림 3]는 배경 이 되는 그림 위로 xml의 정보를 토대로 3개의 AniTextView가 삽입 되어져 있으며, 텍스트뷰의 위치는 개발자에 의하여 조정 가능하다.

화면 하단의 텍스트는 좌측으로 이동하게끔 애니메이션 효과를 준 예제 화면이다.

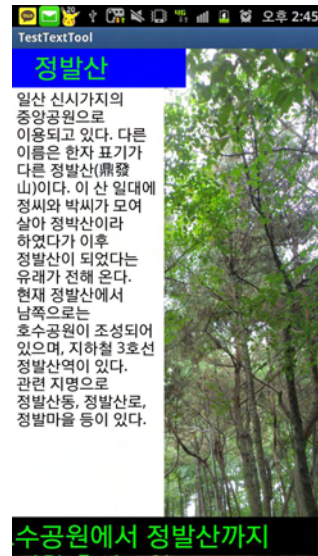


그림 4 안드로이드 화면 구성

## 5. 결론

본 논문에서는 모바일 디바이스 저작도구의 설계를 제안하였으며, 텍스트 편집 기능 및 텍스트 생성을 도와 사용자에게 간편하게 에디터에서 생성하여 모바일 디바이스에 적용하는 저작도구를 설계하였다.

기존의 애플 저작도구의 경우 텍스트만 따로 편집하여 화면에 넣는 것이 아닌 전체 화면을 모두 제작하는 방식이며, 이를 모바일에 넣기 위해서는 무거운 엔진이 필요하지만, 본 논문에서 설계한 저작도구는 각 텍스트뷰를 따로 편집할 수 있게 함으로써 기존의 애플 저작도구보다 엔진이 가벼우며 애니메이션 기능을 특화시켜 보다 특별한 텍스트를 편집할 수 있게 도와준다. 그리고 이 저작도구는 일반적인 모바일 개발 시에도 텍스트를 편집하여 사용 가능하므로 보다 폭넓게 사용할 수 있을 것으로 기대된다.

## 6. 참고

[1] 저작도구(관련 논문)

성동욱, 박준호, 박형순, 노규중, 박 찬, 류관희, 유재수, “데이터 중복을 고려한 웹 기반 UCC 저작 시스템의 설계 및 구현”

[2] 김은수, 박준석, “앱 저작도구를 이용한 교육용 앱개발 연구”

[3] 이정중, 백성욱, “3D 콘텐츠 저작도구와 활용사례”

[4] UCNovel 홈페이지

<http://www.UCNovel.com>

[5] Studio4UX 홈페이지

<http://stydio4ux.comroid.com/reference/android/widget/TextView.html>

[6] Xml 로드

<http://www.gisdeveloper.co.kr/499>

[7] 파워포인트 관련 자료

<http://office.microsoft.com/ko-kr/powerpoint/>

[8] TextView 관련 자료

<http://developer.android.com/reference/android/widget/TextView.html>

[9] SQLite

<http://www.sqlite.org/>