

크로스 플랫폼에서의 탐색 패턴 연구

김희광^{1*}, 박찬익²

¹강남대학교 미래인재개발대학 유니버설디자인전공, ²청운대학교 멀티미디어학과

A study on Cross-Platform Search Pattern

Hwoi-Kwang Kim^{1*}, Chan-Ik Park²

¹Universal Design, Kangnam University

²Multimedia, Chungwoon University

요 약 애플리케이션은 스마트 폰의 발전과 함께 지속적으로 발전되어 왔으며 그와 더불어 탐색구조도 변화하고 있다. 본 연구에서는 크로스 플랫폼의 다양한 사례 분석을 통해 탐색 패턴을 발굴하고 정리하여 그에 따른 올바른 사용 방법을 제시하고자 하였다. 그 결과 크로스 플랫폼을 위한 탐색구조는 지속형 내비게이션, 일시적 내비게이션 그리고 보조적 내비게이션으로 구분할 수 있다. 지속형 내비게이션은 스프링보드, 리스트, 갤러리, 탑 메뉴 등을 사용하여 계층적 구조에서 유효하게 사용이 가능하다. 일시적 내비게이션은 드로워, 토글 메뉴 그리고 파이 메뉴를 활용할 수 있으며 보조 내비게이션은 아코디언 메뉴나 스크롤링 등을 이용하여 크로스 플랫폼에서 동일한 경험을 제공하는 것이 가능하다는 것을 알 수 있었다.

주제어 : 크로스 플랫폼, 탐색구조, 스마트 폰, 지속형 내비게이션, 일시적 내비게이션, 보조 내비게이션

Abstract Applications have evolved with the evolution of smartphones, and the search structure is changing as well. In this study, we searched through various cases of cross platform and found out the search pattern and suggested correct usage method accordingly. As a result, the search structure for cross platform can be divided into persistent navigation, temporary navigation, and auxiliary navigation. Persistent navigation is available in a hierarchical structure using springboards, lists, galleries, and top menus. Temporary navigation can take advantage of drawers, toggle menus, and pie menus, while secondary navigation can provide the same experience across platforms using an accordion menu or scrolling.

Key Words : Cross-platform, Navigation, Smart-phone, Persistent navigation, Temporary navigation, Secondary navigation

1. 서론

2017년 5월 IDC의 스마트 폰 OS마켓의 점유율 조사 결과에 따르면 2016년 이후 Android의 점유율은 줄곧 85%를 차지하고 있으며 애플의 iOS는 14% 전후이고 그 밖에 윈도우 등의 기타 OS가 1%미만인 것으로 나타났다[1]. 2016년 이후 스마트 폰의 OS는 Android의 점유율이 절대적이지만 대표적인 OS를 논할 때 iOS를 포함한다. 스마트 폰은 하드웨어적으로 지속적인 발전을 하

고 있지만 작은 화면이라는 특수한 환경 때문에 그에 맞는 화면을 설계하는 것은 물론 사용자를 배려하는 서비스도 뒷받침 되어야 한다. 스마트 폰에서 가장 중요한 것은 콘텐츠를 탐색 할 수 있는 기본적인 탐색 패턴이라 할 수 있다. iOS와 Android는 지향하는 디자인 콘셉트 뿐 아니라 탐색을 위한 온 스크린 인터페이스를 구성하는 기본 단위인 컴포넌트의 명칭이나 사용법도 다르다. 또 바, 뷰, 컨트롤로 구성된 iOS와 최 상위 수준 뷰, 카테고리 뷰, 세부 뷰로 구분되는 Android는 구조자체가 달라 탐색

*Corresponding Author : Hwoi-Kwang Kim(udekim@kangnam.ac.kr)

Received February 12, 2018

Accepted April 20, 2018

Revised March 14, 2018

Published April 28, 2018

패턴이 다를 수밖에 없다. 또 최근에는 iOS와 Android 모두에서 무리 없이 실행되는 크로스 플랫폼 베이스의 반응형 웹이 네이티브 앱보다 더 활발하게 개발되고 있는 실정이다. 최근의 앱은 네이티브 앱이든 반응형 웹이든 그 종류에 상관없이 99%의 OS점유율을 가지고 있는 iOS와 Android의 두 디바이스에서 일관성을 가진 콘텐츠를 제공하고자 노력하고 있다. 따라서 본고에서는 iOS와 Android에서 공통으로 사용되는 기본적인 탐색구조를 알아보고 크로스 플랫폼에서의 탐색 패턴을 제안해보고자 한다.

2. 애플리케이션의 변화와 탐색구조

초창기의 스마트 폰은 무선 인터넷의 접속은 물론 아이콘을 터치 하는 것으로 애플리케이션을 실행하고 휴대가 간편한 점 등을 이유로 사용자들에게 더욱 친숙한 디바이스로 등장하게 되었다. 앱이라고 불리는 애플리케이션은 OS별로 별도의 플랫폼에 따라 제작을 하였기 때문에 초도 비용은 물론 플랫폼에 따라 다른 UX와 UI의 일관성을 유지하기 위해 노력했다[2]. 초기의 앱은 사용자들에 있어 시행착오를 거치게 만들었지만 비슷한 문제가 발생하였을 때 이전의 경험을 바탕으로 해결법을 응용하고 문제를 손쉽게 해결하게 되면서 스마트 폰에서는 고정적으로 사용하는 패턴이 생겨나게 되었다[3]. 이러한 패턴은 스마트 폰의 사용에 있어 발생할 수 있는 공통적인 문제를 해결하는 솔루션이 될 수 있으며 문제를 해결하는 최상의 가이드가 되기도 한다. 패턴이 생겨나면서 그에 맞는 가이드는 OS별로 업데이트 되면서 제조사를 통해 제공되고 있다. 애플리케이션은 네이티브 앱, 웹 앱, 하이브리드 앱, 모바일 웹으로 구분할 수 있다[4]. 네이티브 앱은 OS별로 플랫폼에 맞는 앱을 말하며 특정 플랫폼에서만 작동하기 때문에 다른 OS에서는 구동되지 않는다. 또 디바이스 별 앱 스토어에서 사용자가 선별적으로 다운로드 받아 설치하게 되는 특징이 있다. 스마트 폰의 카메라 GPS 등과 같은 기능의 활용이 가능하고 OS와 플랫폼에 맞게 제작되어 빠른 속도로 구동되지만 비용이 많이 들며 개발 기간도 장시간 소요된다. 웹 앱은 인터넷 브라우저를 통해 실행되는 앱으로 사용자가 구동되는 브라우저를 선택할 수 있다. 수정 보완이 신속하게 가능하지만 무선 인터넷의 속도에 따라 구동 속도는 다르며 웹의 장점인 쉬운 제작과 네이티브 앱의 구동 방식을 모방

하여 제작된다. 겉모습은 네이티브 앱과 비슷하지만 웹의 속성을 가지고 있기 때문에 디바이스 자체의 하드웨어 기술의 구현은 불가능하다. 하이브리드 앱은 웹과 네이티브 앱의 장점을 결합하여 디바이스가 제공하는 기술의 구현이 가능하고 네이티브 앱 보다 저렴하게 제작이 가능하다는 특성을 가지고 있다. 모바일 웹은 스마트폰의 화면에 맞게 PC의 웹을 그대로 옮겨 온 것을 말한다. 2014년 HTML5의 전 세계 표준 채택으로 인한 반응형 웹이 스마트폰으로 들어오게 된 계기이기도 하다[5-7]. 웹에서 제공하던 동량의 정보를 스마트폰에서 하이브리드 앱이나 웹 앱을 통해 제공하기에는 무리가 있었다. 그로 인해 HTML5를 기반으로 웹의 콘텐츠를 스마트폰으로 옮겨 오게 되었으며 태블릿이나 노트북과 같은 디바이스에서 일관성을 유지하기 위해 반응형 웹을 도입하게 되었다. 반응형 웹의 개발 관계도는 다음 Fig. 1과 같다.



Fig. 1. Sample of responsive web development[8]

반응형 웹은 크기가 다른 여러 디바이스를 고려해 그에 맞는 화면을 제공할 수 있고 웹 앱이나 하이브리드 앱에 모두 담기 어려운 웹과 유사한 수준의 콘텐츠 제공이 가능하며 일관성 유지에 용이하다는 장점으로 인해 2013년 이후 폭발적으로 늘어났다고 할 수 있다[8]. 이러한 대량 콘텐츠의 스마트폰으로의 이동은 콘텐츠의 설계과정에서 5개 이상의 카테고리를 생성할 수밖에 없게 되었으며 이로 인해 드로워 또는 슬라이더라는 UX 패턴이 생겨나게 되었고 그로 인해 수많은 논란이 발생되기 시작했다. 드로워 또는 슬라이더 UX와 함께 필연적으로 사용되는 것은 햄버거 버튼이라고 불리는 아이콘으로부터 촉발되었다고 해도 과언이 아니다. 햄버거 버튼은 아이콘의 모양이 햄버거와 닮았다고 하여 붙여진 이름으로 30여년 전 초기 GUI를 개발한 제록스의 개인용 워크스테이션을 위해 롱 콕스에 의해 개발되었다. 이후 아이폰의 녹음기 앱에서 다시 등장했다가 페이스 북을 통해 유명세를 타게 되었다[9, 10]. 드로워 또는 슬라이더는 햄버거 버튼

을 탭 하여 사용하기 때문에 들은 때려야 뗄 수 없는 관계이지만 작은 화면의 스마트 폰에 대량의 콘텐츠를 그루핑하여 카테고리를 배치하기에 적합한 UI로 논란의 여지와 상관없이 여전히 사용되고 있다. 햄버거 버튼의 논란은 대안을 찾기 위해 많은 개발자와 학자들의 논쟁거리가 되었으며 그로 인해 다른 탐색구조에 대한 논의가 촉발되었다고 할 수 있다.

3. iOS와 Android의 구조적 차이

iOS는 바Bar, 뷰Views, 컨트롤Controls로 구성되어 뷰와 뷰 컨트롤러를 조합한 탐색구조를 가지게 된다. 기술적으로 보면 UI 요소는 UI 뷰UI View 클래스를 상속하므로 일종의 뷰View로 볼 수 있다. 뷰는 화면에 자신을 어떻게 그릴지 알고 있으며, 사용자가 자신의 영역 안에 터치하는 것을 알 수 있다. 앱에서 사용하는 뷰들과 그 계층구조를 관리하기 위해 일반적으로 뷰 컨트롤러를 사용한다. 뷰 컨트롤러는 뷰를 배치해서 보여주고, 사용자 상호작용Interaction에 반응하는 기능을 구현하거나 한 화면에서 다른 화면으로 전환되는 효과를 관리한다. 반면 Android는 플랫폼과 장치에 걸쳐 시각, 운동, 상호작용 설계를 위한 머티리얼 디자인Material Design을 바탕으로 최 상위 수준의 뷰와 세부 편집 뷰로 구성된다. 계층구조가 복잡해지고 이동 경로가 깊어지면 카테고리 뷰가 최 상위 수준과 세부 뷰를 연결하는 구조를 가지고 있다. 최 상위 수준의 뷰는 앱이 지원하는 다양한 뷰로 구성이 되며 각 뷰는 같은 데이터를 어떻게 표현하는지와 앱이 어떠한 기능을 표현하는지에 따라 다양한 모양을 가진다. 카테고리 뷰는 특정 데이터를 세부 뷰와 최상위 뷰로 연결하는 구조를 가지게 되며, 세부 편집 뷰에서는 데이터를 편집하여 보여준다. 이렇듯 iOS와 Android는 각각 다른 구조를 가지고 UI 구성 요소의 명칭을 달리 하고 있으며 그에 따라 사용하는 방법에 대한 가이드라인도 다르다. iOS는 각 구성요소를 바, 뷰, 컨트롤로 구분하고 있으며 Android에서는 그와 관련된 요소들을 모두 컴포넌트와 패턴으로 구분하여 가이드를 제공하고 있다. 그 중에서도 iOS와 Android에서 가장 많이 사용되면서 중복되는 역할을 하는 요소들은 대부분 바Bar, 뷰Views, 컨트롤Controls 정도이다.

탐색을 위해 일반적으로 Android는 기기의 하단에 이

전화면으로 돌아가 사용할 수 있는 버튼이 물리적으로 배치되어 있었지만 최근의 기기들에서는 소프트웨어로 화면 안에 삽입되어 뒤로 가기 버튼의 역할을 대신한다. 반면 iOS에서는 이 버튼이 없기 때문에 이전 화면으로 돌아가기 위해서 일반적으로 화면 왼쪽 상단에 있는 뒤로 가기 버튼을 이용해 앱의 다양한 여정을 고려한다. 아마도 이것이 iOS와 Android의 가장 큰 차이이기도 하고 구성요소별 위치나 명칭이 다른 이유이며 두 OS는 다른 내비게이션 UI를 가지게 된 것이라 할 수 있다.

4. IA구조에 따른 탐색

앱을 실행시키면 스플래시splash나 코치마크coach mark 또는 워크 스루walk through를 지나 처음으로 사용자가 진정한 앱의 기능을 마주 하게 된다[11]. 앱 페이지의 디자인에 있어 가장 크게 고민하고 또 고민 되어야 할 부분은 바로 이 때의 내비게이션이라고 할 수 있다. 사용자들은 자연스럽게 친숙한 느낌의 인터페이스를 선호하며 그를 바탕으로 앱을 탐색하게 되는데 아무리 아름다운 디자인이어도 사용자가 원하는 걸 찾기 힘든 방식으로 UI가 구성되어 있다면 기본적인 기능조차 만족시키기 어렵다[12]. 따라서 앱의 UI 설계를 위해서는 OS별 컴포넌트component와 더불어 내비게이션에 대한 이해가 필요하다.

내비게이션 UI는 화면의 이동을 담당하는 UI로 OS 별로 다른 UI를 사용한다. 내비게이션 UI는 정보 구조의 설정에 따라 디자인되어야 하며 사용성이나 중요도에 따라 그 속성을 잘 알도록 디자인되어야 한다[13, 14]. 내비게이션은 정보의 구조에 따라 계층적Hierarchical navigation, 평면적Flat navigation, 콘텐츠나 경험 기반 Content-driven or experience-driven navigation의 세 가지의 패턴으로 구분될 수 있다[15].

계층적 내비게이션은 위계 구조가 존재하기 때문에 사용자가 목적지에 도달할 때까지 하나의 메뉴에 하나의 화면을 제공하며 다른 목적지로 가거나 단계를 되돌리기 위해서는 처음부터 다시 시작하거나 다른 선택을 해야 한다. 주로 리스트 형태로 제시된 메뉴를 탭하여 하위 메뉴로 진입하는 방식의 UI가 사용되며 설정 및 메일에서 사용하는 방법이다. 계층적 내비게이션의 구조는 다음 Fig. 2와 같다

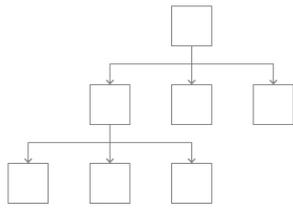


Fig. 2. Hierarchical navigation

평면 내비게이션 Flat navigation은 계층 구조를 가지지 않고 동일한 레벨의 메뉴를 수평적으로 이동하게 된다. 평면의 정보 구조를 바탕으로 메인화면에 항상 메뉴가 보여 지기 때문에 사용자가 하나의 메뉴에서 다른 콘텐츠로 이동하는 것이 즉시 가능하며 주로 탭 바의 형태로 음악 및 앱 스토어에서 사용된다. 평면 내비게이션의 구조는 다음 Fig. 3과 같다.

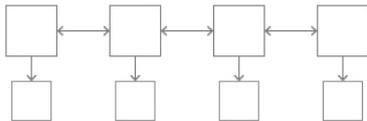


Fig. 3. Flat navigation

마지막으로 콘텐츠나 경험 기반의 내비게이션 Content-driven or experience-driven navigation은 콘텐츠를 통해 자유롭게 이동하거나 콘텐츠 자체가 내비게이션의 역할을 한다. 게임이나 E-book에 적합한 내비게이션이다. 콘텐츠 및 경험 기반 내비게이션의 구조는 다음 Fig. 4와 같다.

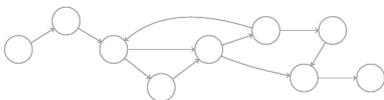


Fig. 4. Content-driven or experience-driven navigation

대표적인 세 가지 내비게이션 외에도 일부 앱들은 복합적인 내비게이션 UI를 제공하는 경우도 있다. 예를 들어 평면 내비게이션의 구조를 가지고 있다고 해도 설정 등의 카테고리 내에서 다른 구조의 내비게이션을 구현하는 것이 가능하다. 내비게이션을 구성할 때는 항상 명확한 경로를 제공하여 사용자들이 앱에서 어디에 있는지 인지할 수 있어야 하며 다음 목적지로 이동하는 방법을 알아야 한다. 내비게이션의 스타일에 관계없이 콘텐츠를

통한 경로는 논리적이고 예측 가능해야 하며 각 화면에 하나의 경로를 제공하는 것이 좋다. 복잡한 상황에서는 경고 창Alert이나 팝업pop-up 또는 모달 뷰modal view를 사용하는 것도 좋은 방법이다. 이러한 내비게이션을 구성하기 위해서는 당연히 선행되어야 하는 것이 정보 구조의 설계이다. 화면에 필요한 방식으로 정보 구조를 구성하여 최소의 탭 수를 사용하게 하는 것이 바람직하다. 또 사용자가 예측 가능한 터치 제스처touch gesture를 사용하여 내비게이션을 예측할 수 있도록 해 주어야 한다. 그 밖에도 탭 바나 컨트롤, 테이블 등의 구성요소를 사용하고 적절한 페이지 컨트롤을 제시해 주어야 한다.

5. 크로스 플랫폼을 위한 탐색 제안

내비게이션의 정보 구조에 따른 계층 구조 이외에도 내비게이션은 항상 그 위치를 준수하는 지속형persist navigation과 일시적으로 나타나는 일시형temporary navigation 그리고 터치 제스처와 같은 보조 내비게이션 supportive navigation으로 나눌 수 있다. 지속적 내비게이션 패턴은 스프링보드, 리스트, 갤러리, 탭 메뉴 등으로 명시적인 메뉴 구조를 표현해 주며 드로어, 토글, 팝업, 모달 뷰 등은 탭 등의 특정 제스처를 통해 접근하는 일시적 내비게이션에 속한다. 스마트 폰이 가진 공간적 크기의 한계 때문에 최근에는 지속형 내비게이션이 아닌 일시적 내비게이션의 사용이 빈번해 지고 있으며 그에 따른 모바일 패턴에 대한 논의가 이루어지고 있는 실정이다.

5.1 지속형 내비게이션

스프링보드는 응용프로그램의 진입을 위해 사용하는 아이콘 모음으로 런치패드 또는 대시보드라고도 부르며 주로 4x5, 4x4, 4x3의 그리드 형태를 가진다. 스프링보드는 크로스 플랫폼에서 잘 작동하기 때문에 일관성과 재사용이 허용되는 OS의 중립 디자인의 사용에 좋다. 개수가 많아지면 한 화면에 표시하는 것이 어렵기 때문에 여러 페이지로 구분되기도 하며 아이콘의 형태를 가진다. 스프링보드의 화면은 꼭 그리드 레이아웃에 따라 일정한 규격의 크기를 가지고 디자인 되는 것은 아니며 중요도에 따라 다른 항목보다 더 크거나 작게 디자인 하는 것도 가능하다. 단점은 모든 옵션을 동일한 중요도로 균등하게 제공하는 것이지만 중요도가 필요한 경우는 윈도우

폰과 같은 타일 구조를 선택하여 크기의 변화에 따른 중요도를 전달할 수 있으며 리스트를 통해 중요도에 따른 계층 구조를 전달할 수 있다. 리스트는 스크롤 할 수 있는 콘텐츠의 목록을 표시하는 요소로 결과 값에 따라 한 줄에서 여러 줄까지 구성이 가능하고 아이콘이나 썸네일 thumbnail 항목을 추가 할 수 있다. 그 밖에도 배지Badge나 토글Toggle 같은 구성 요소를 함께 삽입할 수도 있기 때문에 항목을 추가할 때는 가독성이 떨어지지 않도록 보기 쉽게 배열하는 것이 중요하다. 또 리스트 간의 간격과 조합을 고려하여 규칙적인 화면을 구성되어야 하며 리스트 간 구분선이 사용되기도 한다. 선택한 리스트는 색상이나 투명도를 이용하여 구분하는 것이 좋다. 리스트는 각 항목이 응용 프로그램의 시작 지점이라는 점에서 스프링 보드와 유사하지만 모듈을 전환하려면 목록으로 다시 이동하여 계층적 탐색을 호출하게 된다. 계층형 앱에서 사용자는 목적지에 도달할 때까지 화면 하나당 하나의 선택을 함으로써 탐색을 하게 되는 스마트 폰의 특성상 설정 및 메일과 같은 계층적 구조를 사용하는 것을 크로스 플랫폼에서 유용하게 사용될 수 있다.

갤러리는 탐색을 위한 개별 콘텐츠로 스프링보드와 비슷하게 콘텐츠의 정렬이 바둑판 모양의 표 형태로 구성되는 화면을 말한다. 행과 열로 배치하여 표현하며 이미지의 크기에 따라 같은 크기의 모듈을 반복하여 리스트 업 하거나 이미지의 크기를 다르게 하여 중요도에 따라 배치하기도 한다. 기본적으로는 갤러리나 쇼필름의 리스트 화면에 가장 많이 사용되는 이미지 배열 방식이며, 이미지만 보여주기 보다는 텍스트와 함께 사용하여 사용자의 이해를 돕는 것이 좋다. 갤러리는 계층적 갤러리와 비 계층적 갤러리로 구성될 수 있으며 크로스 플랫폼에서 동일한 경험을 제공하는 것이 가능하다.

탭 메뉴는 가장 많이 사용되는 모바일 UI구성 요소 중 하나로 안드로이드는 상단에, IOS는 하단에 배치되는 특징을 가지고 있다. 대부분 메타포metaphor를 사용한 아이콘을 배치하게 되며 내비게이션 바와 유사하게 연관된 화면 그룹을 보여 줄 때 사용하게 된다. 한 세트의 탭 안에 있는 모든 내용은 서로 관련이 있어야 하며 탭의 내용은 색상이나 음영을 통해 상호 배타적으로 보여준다. 사용자가 탭을 선택 했을 때는 자신이 찾으려는 것을 쉽게 예측 할 수 있도록 논리적인 콘텐츠의 집합으로 묶어 레이블링labelling 하는 것이 좋다. IOS에서는 플랫폼 내비게이션 구조로 IA가 구성된 경우 주 카테고리에서 모든 기

본 카테고리에 액세스가 가능하기 때문에 탭 메뉴를 사용하며 Android의 경우는 계층 구조로 이루어진 슬라이더 내비게이션을 사용하는 차이가 있다. IOS의 탭 바는 메뉴 항목이 5개로 제한되기 때문에 5개 이상의 카테고리가 필요한 경우 다섯 번째 메뉴를 추가 옵션을 제공할 수 있는 아이콘으로 변경하여 사용한다. Android의 슬라이더 내비게이션은 내비게이션 내부에서 상하 스크롤이 가능하고 구분선을 이용하여 카테고리를 배치할 수 있기 때문에 그 사용이 좀 더 자유롭다. 탐색구조에서는 비록 Android가 슬라이더 내비게이션을 기본 패턴으로 채택하고 있지만 탭 바도 기본 패턴에 속한다. 슬라이더 내비게이션과 탭 바의 사용에 있어 차이는 사용자가 자주 사용하는 항목이어야 하며 제한 된 수의 최 상위 레벨보기인 경우 슬라이더 내비게이션 외에 탭 바를 사용하게 된다. 이때 탭 바의 항목은 4개의 고정 탭을 사용하거나 그 이상의 항목이 있는 경우는 스크롤링 탭을 사용하게 된다.

5.2 일시적 내비게이션

드로워는 메뉴가 직접적으로 보이지는 않고 특정 제스처를 통해 접근이 가능하며 주로 리스트 형 메뉴가 오른쪽 또는 왼쪽에서 화면의 절반 이상을 덮으면서 나타난다. 대표적으로 햄버거 메뉴라고 불리는 버튼을 탭 했을 때 반응하며 드로워가 원래의 화면을 덮거나 겹치면서 나타나는 오버레이Overlay 방식과 원래의 화면을 돌출하는 방향으로 밀어 내는 형태로 나타나는 인레이inlay 방식으로 나뉜다. IOS에서는 주로 인레이 드로워를 사용하며 Android 에서는 오버레이 드로워를 사용한다.

토글 메뉴는 사이드 드로워와 비슷하게 인레이나 오버레이 디자인으로 나뉘며 주로 위에서 아래로 내려오거나 메뉴의 일부가 아래로 열리기도 하며 안드로이드에서는 스피너의 형태로 사용이 된다. 토글 메뉴는 메뉴 전체가 화면을 덮어서는 안 되며 백그라운드에서 아무 곳이나 누르면 메뉴가 숨겨진다. 파이메뉴는 원형 메뉴 또는 방사형 메뉴라고 하는데 주로 게임의 세부 메뉴에 활용되는 형태이다. 파이 메뉴는 최근 안드로이드의 오픈소스로 제공되고 있는데 표준으로 채택이 되지 않는 한 OS 간의 충돌로 앱 내 기본 내비게이션으로 사용되기는 어려울 듯하다. 파이 메뉴는 다중 계층을 가진 경우도 있으며 탭하고 누르면 가변적인 크기로 나타나게 되며 최근에는 트리거trigger를 사용하여 나타나게 하는 경우가 많다.

5.3 보조 내비게이션

스와이프 제스처를 사용하여 콘텐츠를 빠르게 탐색하는데 사용할 수 있는 페이지 스와이핑은 전체 페이지가 스와이프 되는 형태와 일부분만 스와이프 되는 디자인으로 구분된다. 전체 페이지가 스와이프 되는 경우는 페이지 표시가 없이 왼쪽 혹은 오른쪽으로 화면이 전환되며 일부분만 사용하는 경우는 페이지의 숫자를 표시하거나 작은 점의 형태로 콘텐츠의 존재를 알려준다.

스크롤링 탭은 특정 모듈 내에서 여러 범주 보기를 할 때 사용한다. 스크롤링 탭은 터치 대상이 아니기 때문에 일반적으로 표준 탭 막대보다 얇으며 가로 방향으로 스와이프 하는 수단으로 사용된다.

아코디언은 동일한 화면 내에서 더 많은 정보를 볼 수 있는 역할을 하며 새 화면으로 이동해서 다시 탐색하는 방법 보다는 효율성을 제공한다. 아코디언은 열리고 닫히는 형태로 이루어져 있기 때문에 화살표의 방향이나 더하기 빼기 등의 부호를 사용해 패널의 확장과 축소 상태에 대한 팁을 제공해 주어야 한다.

6. 결론

IA의 구조에 따른 탐색이나 위치 준수에 따른 탐색은 iOS와 Android 모두에 공통적으로 사용할 수 있는 탐색 방식이다. 하지만 세부적으로 들여다보면 OS의 구조에 따라 약간 다른 방식으로 사용되어야 한다는 것을 알 수 있다. 예를 들면 일시적 내비게이션의 드로워는 안드로이드에서는 대표적인 탐색 구조 방식이라 할 수 있지만 iOS에서는 구조적으로 OS와 충돌을 일으킬 수밖에 없다. 그와 같은 연유로 햄버거 버튼은 가장 사용해서는 안 되는 아이콘이라는 오명을 쓰게 되기도 하였다. 따라서 크로스 플랫폼을 위한 탐색구조를 위해서는 우선 IA의 탐색구조를 먼저 결정할 필요가 있다. 크로스 플랫폼을 위한 탐색구조는 지속형 내비게이션, 일시적 내비게이션 그리고 보조적 내비게이션으로 구분할 수 있다. 지속형 내비게이션은 스프링보드, 리스트, 갤러리, 탭 메뉴 등을 사용하여 계층적 구조에서 유효하게 사용이 가능하다. 일시적 내비게이션은 드로워, 토크 메뉴 그리고 파이 메뉴를 활용할 수 있으며 보조 내비게이션은 아코디언 메뉴나 스크롤링 등을 이용하여 크로스 플랫폼에서 동일한 경험을 제공하는 것이 가능하다는 것을 알 수 있었다. 탐

색구조가 계층을 가지는지 리니어 방식의 플랫폼 구조인지에 따라 지속형 내비게이션을 사용하게 될 것인지 일시적 내비게이션을 사용하게 될 것인지를 결정해야 한다. 만일 계층 구조를 가지고 있는 IA라면 지속적 내비게이션과 일시적 내비게이션을 함께 사용하는 것이 좋다. 본 연구는 최근의 크로스 플랫폼이 점차 많은 비중을 차지하는 바, 그 탐색구조에 관한 기초 자료로서의 활용 가치가 있기를 기대해 본다.

REFERENCES

- [1] *Worldwide smartphone OS Market Share* <https://www.telephonesonline.co.uk/blog/mobile-market-shares/>
- [2] J. Y. Kim. (2012). *A study on smart phones uses and flow formation factors : from the perspective of user experience design factors*. Ihwa womens University, Ph.D. thesis, Seoul.
- [3] J. D. Cho. (2015). *Creative Convergence Capstone Design with PBL*, Seoul. Hanbit Academy.
- [4] S. Y. Shin & K. S. Jeong. (2015). *A Study on Getting the Basic Principles to Apply Safety Design in Public Space -Focused on the Analysis of Public Space Design Guidelines of 4 Cities in Domestic*. *Korea Design Culture*, .21, 281-286.
- [5] Paul Beynon-Davies. (2016). Reeva Lederman, Making sense of visual management through affordance theory, *Production Planning & Control*, 28(02), 144-152.
- [6] N. Kano. (1984). Attractive Quality and Must-be Quality. *Journal of the Japanese Society for Quality Control*. 14(2), 147-156.
- [7] S. J. Yoo, E. B. Choi & H. J. Kim. (2006). An empirical study on the flow experience affected by characteristics of mobile internet. *Information systems review*, 8(1), 125-139.
- [8] *What is responsive web design?* <https://www.mockplus.com/blog/post/pros-and-cons-of-responsive-web-design>
- [9] D. I Hoffman & T. P. Novak. (1996). Marketing in Hypermedia CMEs: Conceptual Foundations. *Journal of Marketing*, 60(3), 50-68.
- [10] H. K. Kim. (2008). *Q Methodology Science philosophy, theory, analysis and application*. Seoul. Communication books Publishing.
- [11] S. R Brown. (1980). *Political subjectivity: Applications*

- of Q methodology in political science*, Newyork. Yale University Press.
- [12] D. I Hoffman & T. P. Novak. (1994). *How Big is the Internet?*, Newyork. HotWired.
- [13] S. U. Pi & M. S. Lee. (2016). An Exploratory Study on the User Experience of Augmented Reality Advertising. *Journal of digital convergence* 14(8), 177-183.
- [14] J. Y. Sung. (2016). Developing convergent class model of augmented reality and ar. *Journal of digital convergence*, 14(5), 85-93.
- [15] D. H. Kim & M. H. Kim. (2015). Design of Mixed reality based edutainment system using cloud service. *Journal of the Korea Convergence Society*, 6(3), 103-109.

김 회 광(Kim, Hwoi Kwang) [정회원]



- 2002년 2월 : 한양대학교 산업디자인(미술학석사)
- 2010년 8월 : 한양대학교 산업디자인(디자인학박사)
- 2012년 9월 ~ 현재 : 강남대학교 미래인재개발대학 유니버설디자인

인전공 교수

- 관심분야 : Design, Marketing
- E-Mail : udekim@kangnam.ac.kr

박 찬 익(Park, Chan Ik) [정회원]



- 1987년 2월 : 서울산업대학교 시각디자인(미술학학사)
- 1994년 2월 : 홍익대학교 산업디자인(미술학석사)
- 2000년 3월 ~ 현재 : 청운대학교 멀티미디어학과 교수

- 관심분야 : VFX, VR, AR, 3D Animation, CG
- E-Mail : butterfly9@chungwoon.ac.kr