

미래창조과학부고시 제2013-107호

국가정보화기본법 제32조제5항의 규정에 의하여 「모바일 애플리케이션 접근성 지침(행정안전부고시 제2011-38호, 2011. 9. 22)」 일부를 개정하고, 다음과 같이 고시합니다.

2013년 8월 19일
미래창조과학부장관

모바일 애플리케이션 접근성 지침

제1장 총 칙

제1조(목적) 이 지침은 「국가정보화기본법」 제32조제5항에 따라 모바일 애플리케이션 서비스 제공자가 장애인과 고령자 등의 접근성을 보장하기 위해 애플리케이션 제작 시 지켜야 할 사항을 규정함을 목적으로 한다.

제2조(용어 정의) 이 지침에서 사용하는 용어의 뜻은 다음과 같으며, 정의하지 아니한 용어는 「국가정보화기본법」 및 동법시행령을 따른다.

1. “접근성”이란 모바일 기기를 사용하여 모바일 애플리케이션을 이용하고자 하는 장애인, 고령자 등을 포함한 모든 사람들에게 이의 활용 가능성이 제공됨을 말한다.
2. “모바일 기기”란 무선 인터넷 서비스를 제공 받을 때에 사용하는 휴대용 기기를 말한다.
3. “모바일 애플리케이션”이란 사용자가 특정한 목적을 달성하기 위하여 모바일 기기 상에서 실행되는 소프트웨어를 말한다.
4. “서비스 제공자”란 모바일 기기를 이용하여 콘텐츠 및 서비스를 제공하는 공공기관 및 사업자를 말한다.
5. “무리한 부담(Undue Burden)”이란 현재 가능한 기술 수준과 적절한 비용으로 실현시킬 수 있는 정도 이상의 노력을 요구함을 말한다.

6. “준수사항”이란 모바일 애플리케이션의 접근성 확보를 위하여 무리한 부담이 되지 않는 한 반드시 준수해야 할 사항을 말한다.
7. “권고사항”이란 모바일 애플리케이션의 접근성 향상을 위하여 무리한 부담이 되지 않는 한 준수할 것을 권장하는 사항을 말한다.
8. “운영체제”란 모바일 기기 상에서 애플리케이션을 비롯한 소프트웨어를 실행할 수 있는 기반 환경을 말한다.
9. “보조기기”란 장애인의 신체적·인지적 기능을 증진, 보완, 향상시키기 위하여 사용하는 기기, 장비의 일부분 또는 시스템, 소프트웨어를 말한다.
10. “터치(touch) 기반”이란 모바일 기기의 기능 및 화면상의 객체를 키나 버튼 같은 물리적인 형태의 컨트롤을 사용하지 않고 화면상에서 직접 만져서 조작 가능한 사용자 인터페이스를 제공하는 운영체제의 특성을 말한다.

제3조(적용 범위) ① 이 지침은 다른 법령에서 별도로 정한 경우를 제외하고는 국가기관, 지방자치단체 및 공공기관 등(이하 “국가기관 등”이라 한다)이 모바일 애플리케이션을 구축, 운영, 개선 및 유지보수할 경우에 적용한다.

② 이 지침이 적용되는 모바일 기기 대상은 다음과 같다.

1. 운영체제를 갖는 모바일 전화기
2. 운영체제를 갖는 태블릿 기기
3. 운영체제를 갖는 전자책 기기

제4조(모든 운영체제에서의 접근성 보장) ① 국가기관 등의 장은 모바일 애플리케이션을 신규 구축하는 경우 무리한 부담이 없는 한 해당 애플리케이션이 기반하고 있는 모든 운영체제에서 동등하게 접근성이 보장된 서비스를 제공하도록 하여야 한다.

② 국가기관 등의 장은 모바일 애플리케이션을 개선, 유지보수 및 운영하는 경우 무리한 부담이 없는 한 모든 운영체제에서 동등하게 접근성이 보장된 서비스를 제공하도록 노력하여야 한다.

③ 제1항 및 제2항에서 무리한 부담이 없는 운영체제의 종류는 해당 애플리케이션을 구축, 운영, 개선 및 유지보수하는 국가기관 등의 장이 정할 수 있다.

제5조(모든 모바일 기기에서의 접근성 보장) ① 국가기관 등의 장은 모바일 애플리케이션을 신규 구축하는 경우 무리한 부담이 없는 한 해당 애플리케이션이 사용되는 모든 모바일 기기에서 동등하게 접근성이 보장된 서비스를 제공하도록 하여야 한다.

② 국가기관 등의 장은 모바일 애플리케이션을 개선, 유지보수 및 운영하는 경우 무리한 부담이 없는 한 모든 모바일 기기에서 동등하게 접근성이 보장된 서비스를 제공하도록 노력하여야 한다.

③ 제1항 및 제2항에서 무리한 부담이 없는 모바일 기기의 종류는 해당 애플리케이션을 구축, 운영, 개선 및 유지보수하는 국가기관 등의 장이 정할 수 있다.

제2장 모바일 애플리케이션 접근성 준수사항

제6조(대체 텍스트) 텍스트 아닌 콘텐츠는 대체 가능한 텍스트와 함께 제공되어야 한다.

1. 대체 텍스트란 그림 및 이미지, 동영상으로 작성된 멀티미디어 형식의 콘텐츠 내용을 텍스트로 그 의미나 기능을 인식할 수 있도록 제공하는 것을 말한다.
2. 텍스트 아닌 콘텐츠에 대한 대체 텍스트는 그 의미나 기능을 파악할 수 있도록 짧고 명확하게 제공해야 한다.

제7조(초점) 모든 객체에는 초점(focus)이 적용되고, 초점은 순차적으로 이동되어야 한다.

1. 초점은 화면상의 선택된 객체의 내용을 화면 낭독 프로그램 등의 보조 기기를 통해 이용할 수 있도록 도와주는 기능을 말한다.
2. 선택된 객체는 초점이 적용되었다고 하고, 초점은 화면상에서 테두리나 하이라이트로 표시하여 제공되는 것이 바람직하다.
3. 표의 객체에 적용되는 초점은 논리적인 순서로 제공되어야 한다.

제8조(운영체제 접근성 기능 지원) 운영체제가 제공하는 접근성 기능 및 속성이 사용되어야 한다.

1. 운영체제에서 제공하고 있는 접근성 기능 지원이 활용되어야 하며, 다음과 같은 사항을 고려할 수 있다.
 - 키보드 등 외부 디바이스와의 호환성 제공을 위한 API
 - 정보 제공 방법의 다중성 (redundancy)
 - 음성명령 기능의 포함, 고대비, 폰트 등
2. 애플리케이션이 해당 운영체제에서 제공하고 있는 접근성 기능을 변경할 경우, 애플리케이션의 종료와 함께 접근성 기능을 변경 전의 상태로 복원시켜야 한다.
3. 입력 서식은 운영체제에서 제공하는 접근성 속성을 활용하여 사용자가 이해하기 쉽도록 해야 한다.

제9조(누르기 동작 지원) 터치(touch) 기반 모바일 기기의 모든 컨트롤은 누르기 동작으로 제어할 수 있어야 한다.

1. 누르기 동작은 화면상의 객체를 손가락 끝으로 접촉하여 만지거나(touch) 가볍게 두드리는(tap) 동작을 말한다.
2. 두 개의 손가락을 동시에 이용해야 하는 다중 누르기(Multi-touch) 동작은 단순한 누르기 동작으로 대체할 수 있는 방법이 제공되어야 한다.
3. 슬라이드(Slide), 끌기와 놓기(Drag and drop) 등의 복잡한 누르기 동작은 단순한 누르기 동작으로 대체할 수 있는 방법이 제공되어야 한다.

제10조(색에 무관한 인식) 화면에 표시되는 모든 정보는 색에 관계없이 인식할 수 있어야 한다.

1. 색상으로 정보를 구분할 경우, 색상 이외의 다른 방법으로도 동등한 내용을 전달할 수 있도록 설계한다.
2. 색상을 사용한 의미의 전달이 흑백 화면에서도 동등하게 이루어질 수

있도록 제공해야 한다.

제11조(명도 대비) 화면에 표시되는 모든 정보는 전경색과 배경색이 구분될 수 있도록 최소 대비 이상으로 제공되어야 한다.

1. 명도 대비는 화면의 배경색과 객체를 표시하는 데에 사용되는 전경색 사이의 명도 차이의 비율(contrast)을 말한다.
2. 고대비 제공이 불가능할 경우, 애플리케이션의 설정 기능에 명도 대비 조절 기능을 제공한다.
3. 화면상의 모든 정보의 최소 대비는 3:1 이상이어야 한다. 저시력인, 고령자 등에게 실효성을 가지기 위해서는 명도 대비가 4.5:1 이상이 되는 것이 바람직하다. 단, 사진과 동영상은 예외로 한다.

제12조(자막, 수화 등의 제공) 멀티미디어 콘텐츠에는 동등한 내용의 자막, 원고 또는 수화가 제공되어야 한다.

1. 자막, 원고 또는 수화는 화면 상의 콘텐츠와 동기화하여 제공하는 것이 바람직하다.

제3장 모바일 애플리케이션 접근성 권고사항

제13조(기본 사용자 인터페이스 컴포넌트) 운영체제에서 제공하는 기본 사용자 인터페이스 컴포넌트(Native UI Component)를 최대한 이용하는 것이 바람직하다.

1. 운영체제에서 제공하는 접근성 있는 기본 사용자 인터페이스 컴포넌트는 사용자 인터페이스 구성에 사용되는 표준 도구(대화상자, 버튼과 체크박스, 타이틀 바 등)들을 말한다.
2. 운영체제에서 제공하는 기본 사용자 인터페이스 컴포넌트를 활용하면 보조기기와의 호환성을 제공하기 용이하므로 접근성의 확보를 위해 적극적으로 활용되어야 한다.

제14조(컨트롤간 충분한 간격) 컨트롤은 충분한 간격으로 배치하는 것이 바람직하다.

1. 컨트롤은 버튼 또는 위젯과 같이 사용자 인터페이스 화면에서 누르기 동작으로 기능을 활성화시키는 객체를 말한다.
2. 좁은 화면 공간의 경우, 사용자의 의도와 무관하게 다른 컨트롤을 누르게 되는 문제가 발생할 수 있으므로, 이를 피하기 위해서 컨트롤 사이의 공간을 충분히 확보하여 사용자가 컨트롤 영역을 명확히 구분할 수 있도록 하는 것이 바람직하다.
3. 모바일 기기의 화면 크기에 관계없이 컨트롤 중심간 간격은 13mm 이상을 권장한다.

제15조(알림 기능) 사용자에게 알림을 제공할 때에는 진동, 시각, 소리 등 최대한 다양한 방법으로 사용자가 선택할 수 있도록 제공하는 것이 바람직하다.

1. 화면상의 모든 알림 정보는 한 가지 양식으로만 제공되지 않도록 하며, 다양한 감각 양식을 활용한다.
2. 사용자가 자신에게 가장 편리한 방법을 선택할 수 있도록 한다.

제16조(범용 폰트 이용) 폰트의 크기 조절, 확대 기능을 제공하거나 운영 체제에서 제공하는 관련 기능을 활용할 수 있는 방법을 제공하는 것이 바람직하다.

1. 범용 폰트(Global Font)는 운영체제에 내장되어 확대나 축소, 기울임 등의 변형 형태가 제공되는 글자체를 말한다.
2. 모든 애플리케이션 화면에서 폰트 크기의 조절이 가능하도록 설계하거나, 최소한 확대 기능을 제공한다.
3. 폰트 크기 조절을 용이하게 하기 위해서는 텍스트 이미지보다 폰트가 지정되어 있는 텍스트를 사용하는 것이 바람직하다.

제17조(사용자 인터페이스의 일관성) 사용자 인터페이스 요소들의 배치를 일관성 있게 제공하는 것이 바람직하다.

1. 사용자 인터페이스를 구성하고 있는 요소들은 사용자가 다시 학습할 필요가 없도록 해당 애플리케이션 내에서 일관성 있게 설계한다.
2. 애플리케이션의 버전이 바뀌어도 중요한 사용자 인터페이스 요소들의 배치는 일관성을 유지한다.

제18조(깜박거림의 사용 제한) 광과민성 발작을 일으킬 수 있는 콘텐츠를 제공하지 않는 것이 바람직하다.

1. 깜빡이거나 번쩍이는 객체를 사용자 인터페이스에 사용하지 않는다.
2. 화면상에서 반드시 깜빡임의 효과를 제공해야 하는 콘텐츠는 초당 3 ~ 50 회의 주기는 피해서 설계한다.

제19조(배경음 사용 금지) 자동으로 재생되는 배경음을 사용하지 않는 것이 바람직하다.

1. 자동으로 재생되는 동영상, 음악, 음성 안내 등을 사용하지 않는다. 단, 3초 미만의 배경음은 예외로 인정한다.
2. 배경음을 사용할 경우, 사용자가 손쉽게 멈춤, 일시정지, 음량조절 등을 제어할 수 있는 수단을 제공한다.

제20조(장애인 등 사용자 평가) 애플리케이션 개발 시 다양한 모바일 기기에서의 이용 가능 여부를 점검해야 하며, 장애인 사용자 평가를 수행하는 것이 바람직하다.

1. 애플리케이션의 출시 이전에 장애인, 고령자 등의 사용자를 대상으로 한 평가를 수행하도록 한다.
2. 사용자 평가는 무리한 부담이 되지 않는 시각 장애, 청각 장애, 뇌병변 장애, 지적 장애, 지체 장애, 고령 등의 사람들을 대상으로 실시한다.
3. 모바일 애플리케이션 서비스 제공자는 해당 애플리케이션의 장애인 등 사용자 평가의 구체적인 결과를 별도로 공시하는 것이 바람직하다.

제4장 보 칙

제21조(준수지침 사항) 제6조 내지 제20조의 모바일 애플리케이션에 대한 구체적인 준수 지침을 구현하기 위한 방법은 별표 1과 같다.

제22조(콘텐츠 접근성 범위) 애플리케이션이 콘텐츠를 제공하는 경우, 이 지침은 서비스 제공자가 직접 제공하는 콘텐츠에만 적용된다.

제23조(접근성 진단) 미래창조과학부 장관은 국가기관 등을 대상으로 모바일 애플리케이션의 접근성 준수 여부를 연 1회 이상 진단할 수 있다.

부 칙

제1조(시행일) 이 지침은 고시한 날로부터 시행한다.

제2조(재검토기한) 「훈령·예규 등의 발령 및 관리에 관한 규정」(대통령 훈령 제248호)에 따라 이 고시 발령 후의 법령이나 현실 여건의 변화 등을 검토하여 이 고시의 폐지, 개정 등의 조치를 하여야 하는 기한은 2016년 3월 23일까지로 한다.

모바일 애플리케이션 접근성 지침 사례

본 모바일 애플리케이션 접근성 지침 사례는 모바일 애플리케이션 개발자 및 운영자들이 접근성을 고려하여 애플리케이션을 개발할 수 있도록 도움을 주기 위해 제공하는 것이다. 본 지침에 포함된 준수사항 7개, 권고사항 8개에 대한 준수 필요성, 대표적인 기술 구현 방법, 구축 사례를 제공한다.

가. 준수사항(7개) : 장애인이 비장애인과 동등하게 모바일 애플리케이션에 접근하여 이용하기 위해서 반드시 준수해야 하는 사항

1. **(대체 텍스트)** 텍스트 아닌 콘텐츠는 대체 가능한 텍스트를 함께 제공해야 한다.

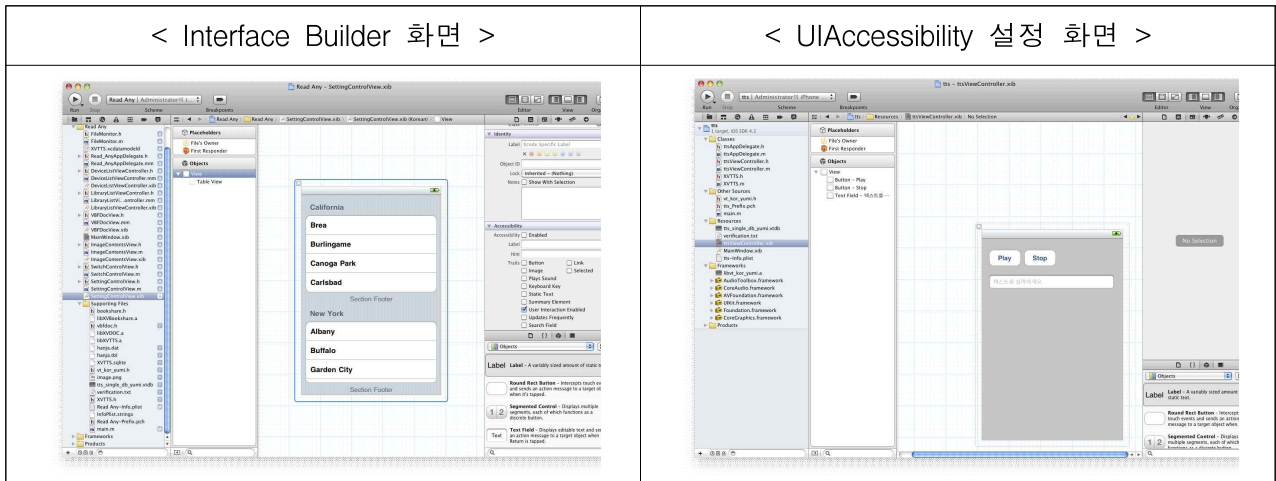
1) 준수 필요성 : 시각장애인의 경우 모바일 기기나 모바일 화면낭독 프로그램에서 제공하는 음성읽기 기능(iOS VoiceOver, 안드로이드 Talkback, 심비안 및 윈도우 모바일 6.5에서 활용되는 Code Factory사의 Mobile Speak 등)을 활용하여 애플리케이션의 정보를 인식할 수 있다. 하지만, 해당 어플리케이션에서 컨트롤 및 객체를 텍스트 아닌 콘텐츠로 제공하면서 대체 텍스트를 함께 제공하지 않으면 시각장애인은 잘못된 정보를 얻거나 전혀 정보를 얻을 수 없다.

2) 기술구현 방법 : 이미지 등 텍스트 아닌 콘텐츠를 제공할 경우에는 반드시 대체 텍스트를 함께 제공해야 한다. 대체 텍스트는 가능한 짧고 명확(Short & Clear)하게 제공하는 것이 바람직하며, VoiceOver, Talkback 등 모바일 스크린리더에서 기본적으로 제공해 주는 용어인 “버튼”, “이미지”, “레이블” 등은 중복해서 제공하지 않는 것이 바람직하다(웹 접근성 연구소 버튼(×), 모바일 접근성 이미지(×), 금융자동화기기 접근성 레이블(×) → 웹 접근성 연구소(○), 모바일 접근성(○), 금융자동화기기 접근성(○)).

① 애플의 iOS에서 대체 텍스트 제공 방법



대체 텍스트를 제공하는 방법은 크게 2가지 방법이 있다. 첫째, 애플에서 제공하는 Interface Builder를 이용하여 대체 텍스트를 제공한다. 둘째, 대체 텍스트를 UIAccessibility API 등을 활용하여 직접 코드에 삽입하여 제공한다. 이러한 경우 Label

과 Hint라는 두 가지의 속성 값을 활용할 수 있는데, Label로 대체 텍스트를 제공하고, Hint는 추가적인 정보가 필요할 경우 제공한다. iPhone에서 사용자가 VoiceOver 옵션에서 Hint를 끌 수(Off) 있으므로 반드시 대체 텍스트는 Label로 제공해야 한다. 예를 들면 뮤직 플레이어의 “멈춤”은 Label에 적고 “뮤직플레이어의 음악을 멈춥니다.”를 Hint에 적는다면 시각장애인 등에게 도움이 될 것이다.



㉠ Interface Builder 이용

Interface Builder는 애플에서 제공하는 것으로 모바일 애플리케이션 개발 시 UI를 손쉽게 만들 수 있도록 도와주는 도구이다. Interface Builder를 실행 시킨 다음 Attribute inspector 창에서 Label 속성에 대체 텍스트를 제공하면 된다. 또한, 애플에서는 접근성 준수여부를 손쉽게 점검할 수 있도록 Accessibility inspector 등을 제공하고 있다.

< Interface Builder를 활용하여 버튼에 대체 텍스트를 제공하는 방법 >		
		<p data-bbox="973 1512 1284 1545"><u>대체 텍스트 제공 방법</u></p> <ol style="list-style-type: none"> <li data-bbox="821 1579 1436 1668">1) Accessibility 속성을 반드시 활성화(Enabled) 시킨다. <li data-bbox="821 1691 1436 1825">2) Label 속성에 텍스트 아닌 콘텐츠의 의미와 정보를 동등하게 인식할 수 있도록 대체 텍스트를 짧고 명확하게 제공한다. <li data-bbox="821 1848 1436 1960">3) 부가적인 설명이 필요할 경우에는 Hint 속성을 활용하여 추가적인 정보를 제공한다.

< 참고 > Interface Builder는 Xcode에 포함되어 제공되고 있다. '11년 7월말 현재 애플의 최신 개발도구는 Xcode 4이다(<http://developer.apple.com/xcode/index.php>).

㉑ UIAccessibility API 등을 활용하여 직접 코드에 삽입하는 경우

애플에서 제공하는 개발자 도구인 Interface Builder를 활용하지 않고 UI를 개발할 경우에는 UIAccessibility API를 활용하여 Label에 대체 텍스트를 제공한다.

```
[houseButton setIsAccessibilityElement:YES];  
[houseButton setAccessibilityLabel:@"멈춤];  
[houseButton setAccessibilityHint:@"뮤직플레이어의 음악을 멈춥니다."];
```

그러나, 보다 효율적인 개발 및 접근성을 제고하기 위해서는 애플에서 제공하는 Interface Builder를 활용하여 개발하는 것이 바람직하다.

㉒ 구글의 안드로이드에서 대체 텍스트 제공 방법

안드로이드의 경우에도 UI에 대체 텍스트를 제공하는 방법은 2가지로 XML을 활용하거나 또는 Java code를 활용하는 방법이 있다. 안드로이드의 경우에는 android.contentDescription 속성을 사용하여 대체 텍스트를 제공해야 한다. "웹 접근성 연구소"라는 버튼에 대체 텍스트를 제공하는 방법은 다음과 같다.

㉑ XML을 활용하여 버튼을 제공할 경우

```
<ImageButton  
    android:id="@+id/add_entry_button"  
    android:src="@drawable/plus"  
    android:contentDescription="@string/add_note"/>>  
layout/main.xml
```

```
<xml    version="1.0" encoding="utf-8"?>  
<resources>  
    <string    name="add_note">웹 접근성 연구소</string>  
</resources>  
values/strings.xml
```

② Java code를 활용하여 버튼을 제공할 경우

```
Button add_entry_button = new Button(this);
add_entry_button.setContentDescription("웹 접근성 연구소");
```

3) 구축 사례

이미지 등 텍스트 아닌 콘텐츠를 활용할 경우에는 반드시 대체 텍스트를 제공해야 한다. 하지만 아래의 ** 은행의 '예금조회/이체' 화면의 사례처럼 대체 텍스트를 제대로 제공하지 않을 경우, 시각장애인 등은 해당 애플리케이션을 이용하기 어렵다. iOS VoiceOver를 설정할 경우, "이전", '전예금조회' '계좌이체' 등의 상단 및 주 메뉴와 '새 소식', '고객센터' 등 하단 메뉴를 모두 "버튼"으로 인식하게 되어 어떤 버튼이 어떤 기능을 수행하는지 알 수 없게 된다. 해당 기능을 제대로 인식할 수 있도록 대체 텍스트를 제공할 필요가 있다.

< ** 은행의 '예금조회/이체' 화면 >	< iOS VoiceOver 설정시 '예금조회/이체 화면' >
	<div data-bbox="587 1081 1007 1706">  </div> <div data-bbox="1034 1081 1401 1294"> <p>대체 텍스트를 미제공하여 '예금조회/이체'의 모든 기능을 제대로 활용하기 어려운 실정(모든 것을 '버튼'이라고 인식)</p> </div> <div data-bbox="1034 1357 1321 1391"> <p>< 상단 및 주 메뉴 ></p> </div> <div data-bbox="1034 1406 1321 1529"> <ul style="list-style-type: none"> - '이전' 버튼 - '전예금조회' 버튼 - '계좌이체' 버튼 등 </div> <div data-bbox="1034 1592 1233 1626"> <p>< 하위 메뉴 ></p> </div> <div data-bbox="1034 1641 1321 1715"> <ul style="list-style-type: none"> - '새 소식' 버튼 - '고객센터' 버튼 등 </div>

텍스트 아닌 콘텐츠에 대한 대체 텍스트 제공 여부에 대해서는 iOS VoiceOver, 안드로이드의 Talkback 등의 음성 읽기 기능을 활성화해 보면 쉽게 제공 여부를 평가해 볼 수 있다.

< 모바일 애플리케이션 접근성 우수 사례 - 청와대 아이폰용 애플리케이션 >

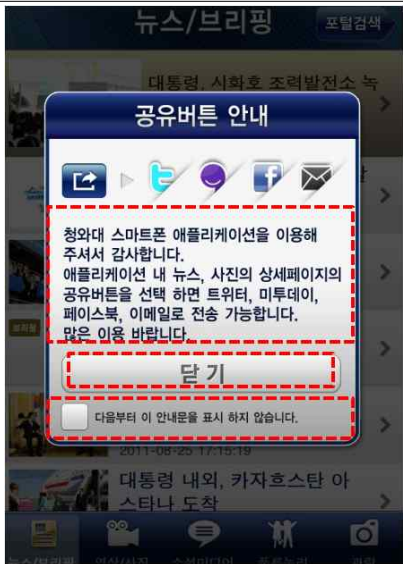
청와대에서는 장애인의 모바일 접근권 보장에 해외 선진사례 분석 및 장애인의 요구에 부응하여, 2011년 8월부터 아이폰 용 애플리케이션의 접근성 제고를 위한 작업을 수행하였다. 청와대에서는 8월 11일과 8월 25일에 2번에 걸친 판올림(update)을 통하여 시각장애인 등의 정보 접근권을 크게 향상시켰다. 국내에서 모바일 애플리케이션 접근성 제고를 위한 우수 사례로 다음과 같은 개선이 이루어졌다.

㉠ 청와대 아이폰 용 애플리케이션의 접근성 관련 판올림 현황(2회)

< 앱 접근성 제고 판올림 1차(8.11) >	< 앱 접근성 제고 판올림 2차(8.25) >
	


㉡ 청와대 모바일 애플리케이션의 개선 전과 개선 후의 비교

보이스 오버를 활용하여 음성을 출력할 경우 대체 텍스트를 제공하지 않아, 청와대 모바일 애플리케이션의 첫 실행 화면의 정보를 얻을 수 없었으나, 대체 텍스트를 제공하여 동등한 정보를 인식할 수 있게 개선됨.


< 공유버튼 안내 페이지 >	< 접근성 개선 전 >
	<p>a. 공유버튼 설명문은 “이미지”로 출력 b. 안내문 버튼 설명 “이미지” “입력 창”으로 출력 c. 닫기 버튼은 “버튼”이라고 출력</p>
	<p>< 접근성 개선 후 ></p> <p>a. 공유버튼 설명문의 내용을 동등하게 음성으로 출력 : <u>청와대 스마트폰 애플리케이션을 ~ (중략)</u> b. 안내문 버튼 설명을 동등하게 음성으로 출력 : <u>다음부터 이 안내문을 ~ (중략)</u> c. 닫기 버튼 : <u>“닫기 버튼”이라고 음성 출력</u></p>

< 모바일 애플리케이션 접근성 우수 사례 - 청와대 아이폰용 애플리케이션(계속) >

보이스 오버를 활용하여 음성을 출력할 경우 하단의 버튼에 대한 대체 텍스트를 제공하지 않았으나, 대체 텍스트를 제공하여 동등한 정보를 인식할 수 있게 개선됨.

< 하단 버튼 >	< 접근성 개선 전 >
	<p>a. 뉴스/브리핑 : "버튼"이라고 출력 b. 영상/사진 : "버튼"이라고 출력 c. 소셜미디어 : "버튼"이라고 출력 d. 푸른누리 : "버튼"이라고 출력 e. 관람 : "버튼"이라고 출력</p>
< 접근성 개선 후 >	
	<p>a. 뉴스/브리핑 : "<u>뉴스, 브리핑 버튼</u>"이라고 출력 b. 영상/사진 : "<u>영상, 사진버튼</u>"이라고 출력 c. 소셜미디어 : "<u>소셜 미디어 버튼</u>"이라고 출력 d. 푸른누리 : "<u>푸른누리 버튼</u>"이라고 출력 e. 관람 : "<u>관람 버튼</u>"이라고 출력</p>

보이스 오버를 활용하여 음성을 출력할 경우 대체 텍스트를 제공하지 않아, 청와대 관람 정보를 얻을 수 없었으나, 대체 텍스트를 제공하여 동등한 정보를 인식할 수 있게 개선됨.

< 이미지 콘텐츠 영역 >	< 접근성 개선 전 >
	<p>본문의 내용인 관람운영일, 관람운영일 설명, 신청대상, 신청대상 설명 등을 파악할 수 없음 : "*** (파일명) 이미지"라고 출력</p>
< 접근성 개선 후 >	
	<p>< 이미지에 대한 동등한 내용의 대체 텍스트 제공 ></p> <p>a. "<u>관람운영일</u>", "<u>매주 화요일 ~ 금요일 (둘째주 토요일)</u> 토요일은 10인 이하의 개인/가족에 한함"으로 음성 출력</p> <p>b. "<u>신청대상</u>", "<u>초등학생 이상 → 미취학자녀 관람은 가족 동반만 가능</u>"으로 음성 출력</p>



2. (초점) 모든 객체에는 초점(focus)이 적용되고, 초점은 순차적으로 이동되어야 한다.

1) 준수 필요성 : 초점이 적용되지 않으면 모바일 기기나 모바일 화면낭독 프로그램에서 제공하는 음성 읽기 기능(iOS VoiceOver, 안드로이드 Talkback, 심비안 및 윈도우 모바일 6.5에서 활용되는 Code Factory사의 Mobile Speak 등) 등을 활용하는 경우의 사용자는 정보를 얻거나 기능을 실행할 수 없는 문제가 발생한다. 또한 순차적으로 이동되지 않을 경우에는 시각장애인, 지적장애인 등이 해당 애플리케이션의 기능과 정보의 이해에 어려움이 발생할 수 있다.

2) 기술구현 방법 : 애플리케이션 개발 시 모든 객체에 초점이 적용되고, 적용된 초점은 순차적으로 이동할 수 있게 개발해야 한다.

① 애플의 iOS에서 초점 제공 방법

Accessibility 속성이 활성화(Enabled)될 경우에만 해당 객체에 초점이 적용된다. iOS에서 제공하는 기본 사용자 인터페이스 컴포넌트(Native UI Component)에서는 대부분의 기본 값이 활성화되어 제공된다. 하지만 UIImageView UI 컴포넌트의 경우에는 Accessibility 속성이 비활성화(Disabled)되어 있다. 만약 단순한 장식이 아니고 의미를 포함하는 이미지를 UIImageView UI 컴포넌트를 활용하여 제공할 경우에는 Accessibility 속성을 활성화 시켜야 하며, 대체 텍스트를 label 속성으로 반드시 제공해야 한다. iOS 4.1부터 초점의 순서를 제어 할 수 있는 기능을 제공하고 있다.

< Accessibility 속성이 활성화 되지 않은 경우 (초점 미적용) >	< Accessibility 속성이 활성화 된 경우 (초점 적용) >
 <p>The screenshot shows the Xcode interface with the Accessibility inspector open. The 'Accessibility' section is expanded, and the 'Accessibility' checkbox is unchecked, indicating it is disabled. Below this, the 'Label' field is empty, and the 'Hint' field is also empty. Under the 'Traits' section, 'Image' is checked, and 'User Interaction Enabled' is also checked.</p>	 <p>The screenshot shows the Xcode interface with the Accessibility inspector open. The 'Accessibility' section is expanded, and the 'Accessibility' checkbox is checked, indicating it is enabled. Below this, the 'Label' field is empty, and the 'Hint' field is also empty. Under the 'Traits' section, 'Image' is checked, and 'User Interaction Enabled' is also checked.</p>

② 구글의 안드로이드에서 초점 제공 방법


안드로이드에서 제공하는 기본 사용자 인터페이스 컴포넌트(Native UI Component)에서는 iOS와 달리 대부분의 기본 값이 비활성화(false)되어 제공된다. 그러므로 초점 이동이 필요한 UI의 경우에는 반드시 focusable 속성을 활성화(true)시켜 제공해야 한다.


```
<TextView android:id="@+id/text"
    android:focusable="true"
    android:text="Hello, I am a focusable TextView"
    android:nextFocusUp="@id/edit"
    ... />
```


안드로이드에서 초점의 이동 순서를 제어하기 위해서는 nextFocusDown, nextFocusUp, nextFocusRight, nextFocusLeft 속성을 이용하면 된다.

3) 구축 사례

모든 컨트롤에 대하여 초점이 적용되어 있어야 하며, 논리적인 순서로 제공되어야만 시각장애인 등이 활용할 수 있다.

< ** 은행의 로그인 화면 >	< 문제점 및 해결방안 >
	<p>o 문제점</p> <ul style="list-style-type: none"> - VoiceOver를 실행할 경우, ‘공인인증서 로그인’, ‘비밀번호 입력창’, ‘로그인’ 등에 초점이 가지 않아 애플리케이션 이용이 불가능한 실정 * 비밀번호 입력 등이 불가능하여 로그인을 할 수 있는 방법이 없음 - ‘공인인증서 로그인’, ‘조회회원 가입’, ‘로그인’ 등에 대체 텍스트도 미제공 * VoiceOver를 사용할 경우, ‘공인인증서 로그인’의 정보가 “dtm into 1 gif”로 출력되는 문제 발생 <p>o 해결방안</p> <ul style="list-style-type: none"> - 모든 버튼 및 정보에 초점이 적용될 수 있도록 제공해야 한다. - 모든 텍스트 아닌 콘텐츠에는 대체 텍스트를 제공해야 한다.

< ** 은행의 조회 화면 >	< 문제점 및 해결방안 >
	<p>○ 문제점</p> <ul style="list-style-type: none"> - ‘조회’ 라는 좌우로 스크롤 되는 메뉴가 나타나는데 이 부분에서는 초점이 전혀 생성되지 않는다. 이 메뉴를 통해 ‘조회/이체’ 등을 선택한 후 하단에서 하위 메뉴로 진입해야 하는데 메인 메뉴에서 초점이 생성되지 않아 더 이상의 은행업무 수행을 하는 것은 불가능함 <p>○ 해결방안</p> <ul style="list-style-type: none"> - ‘조회’ 라는 메뉴에 초점이 적용될 수 있도록 제공하고, 하위 메뉴로 초점이 논리적으로 이동할 수 있도록 제공할 필요가 있음

< ** 방송국의 라디오 >	< 문제점 및 해결방안 >
	<p>○ 문제점</p> <ul style="list-style-type: none"> - *** 월드 라디오는 크게 프로그램과 뮤직채널로 나뉘어져 있다. 이 애플리케이션을 실행하면 두 채널 중 하나를 선택해야 청취가 가능한데, 가운데 있는 이 부분이 VoiceOver를 통해서는 초점이 생성되지도 동작하지도 않는다. 선택하지 않으면 라디오를 들을 수 없는 상황이 되기 때문에 단 하나의 컨트롤의 문제라고 할 수도 있겠지만 VoiceOver 사용자는 이 애플리케이션을 사용할 수 있느냐 사용하지 못하느냐를 가늠하는 중요한 열쇠가 된다. 결과적으로 접근할 수가 없으니 이 애플리케이션의 라디오 기능을 사용할 수 없다. <p>○ 해결방안</p> <ul style="list-style-type: none"> - 시각장애인 등이 이용할 수 있도록 “프로그램”과 “뮤직채널”에 초점을 적용해야 한다.

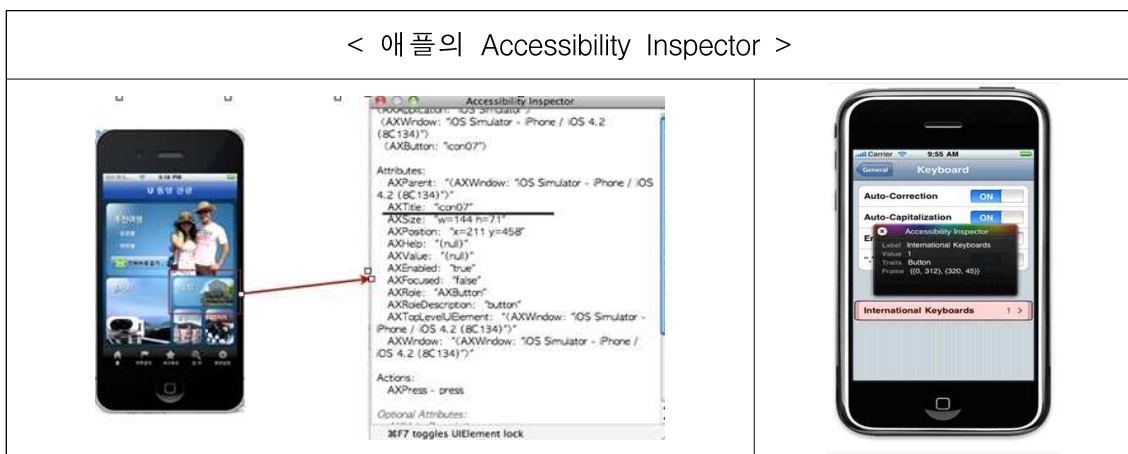
3. (운영체제 접근성 기능 지원) 운영체제가 제공하는 접근성 기능 및 속성이 지원되어야 한다.

1) 준수 필요성 : 장애인 등의 모바일 애플리케이션 접근성을 보장하기 위해서는 각 운영체제에서 제공하는 접근성 기능을 활용해야 한다. 이를 활용하지 않을 경우에는 장애인이 활용하는 다양한 모바일 기기, 모바일 보조기기 등과 호환되지 않아 이용이 불가능할 경우가 발생할 수 있다.

2) 기술구현 방법 : 애플의 iOS와 구글의 안드로이드 운영체제의 경우 각각 접근성 API를 제공하고 있다. 개발자는 각 운영체제에서 지원하는 운영체제의 접근성 기능을 최대한 지원하도록 노력해야 한다.

① 애플의 iOS에서 운영체제 접근성 기능 제공 방법

iOS에서는 Accessibility API를 제공 하고 있다. 모바일 애플리케이션 개발 시 접근성 준수 여부를 점검할 수 있도록 Accessibility Inspector와 Accessibility Verifier 같은 도구들도 제공하고 있다. 다만, 이러한 기능은 매킨토시 PC 환경에서만 이용이 가능하다.



iOS에서 제공하는 UI Accessibility 속성은 5가지이다. 첫째, Label 속성은 텍스트 아닌 콘텐츠에 대한 동등한 정보를 제공하기 위해 사용할 때 이용한다. HTML에서의 alt="" 속성과 유사한 것으로 이미지 등의 텍스트 아닌 콘텐츠 제공시에는 반드시 제공해야 하는 속성이다. 둘째, Traits 속성은 객체의 형태를 표시하는 속성으로 상태, 액션 등을 설명할 수 있으며 다중으로 선택이 가능하다. 현재 Trait 속성 값은 Button, Link, Search Field, Keyboard Key, Static Text, Image, Plays Sound, Selected, Summary

Element, Updates Frequently, Not Enabled, None 등이 포함된다. 셋째, Hint 속성은 부가적인 설명이 필요할 경우에만 사용하는 것으로, Label 등으로 정보가 불충분할 경우 사용하는 속성이다. 넷째, Frame 속성은 화면의 좌표 위치를 나타내 준다. 마지막으로 Value 속성은 각 객체의 속성 값을 말해준다.

② 구글의 안드로이드에서 운영체제 접근성 기능 제공 방법

안드로이드에서도 iOS와 마찬가지로 Accessibility API를 제공하고 있다. Accessibility Service, Accessibility Event, Accessibility API for Customize, Eyes-free 등이 대표적이다. 특히 Eyes-free API는 오픈 소스로 시각장애인 등이 안드로이드용 모바일 기기를 활용할 수 있도록 Talkback, Soundback, Kickback 등을 제공하고 있다.

3) 구축 사례

운영체제에서의 접근성 기능을 지원한 Twitter의 경우 SNS에서 제공하는 텍스트 정보를 제대로 인식할 수 있으나, 국내의 모 SNS의 경우에는 운영체제의 접근성 기능을 지원하지 않아 VoiceOver 기능 이용시 아무런 텍스트 정보도 얻지 못하는 문제가 발생하고 있다.

< Twitter 애플리케이션(운영체제 접근성 지원) >	< **** 애플리케이션(운영체제 접근성 미지원) >
	

4. (누르기 동작 지원) 모든 컨트롤은 누르기(touch or tap) 동작으로 제어할 수 있어야 한다.

1) 준수 필요성 : 시각장애인의 경우 모바일 기기에서 가장 어려움을 많이 겪는 부분이 바로 컨트롤의 위치에 대한 부분이다. 특히 컨트롤을 이동해야 하는 경우나 컨트롤 간의 위치를 바꾸어야 하는 경우 더욱 더 어려움에 처하게 된다. 예를 들어 모바일 애플리케이션을 이동하여 폴더를 생성하는 경우나 전화 받기 기능 같은 경우 Drag해야 해당 기능을 활용할 수 있는 경우가 있다. 그러므로 모든 컨트롤이나 Drag 등이 필요한 기능의 경우에는 반드시 이를 대체할 수 있는 수단인 누르기(touch or tap) 동작을 함께 제공해야 한다.

2) 기술구현 방법 : 누르기 동작은 화면 상의 객체를 손가락 끝으로 접촉하여 만지거나(touch) 가볍게 두드리는(tap) 동작을 말하는 것으로, 모든 컨트롤 동작은 누르기 동작만으로도 제어할 수 있어야 한다. 두 개의 손가락을 동시에 이용해야 하는 다중누르기(Multi-touch) 동작이나 Slide, Drag and drop 등의 복잡한 누르기 동작은 단순한 누르기 동작으로 대체할 수 있는 방법이 제공되어야 한다.

3) 구축 사례 : 모든 컨트롤은 누르기 동작으로 제어할 수 있도록 제공해야 한다. 사용자 편의성 제고를 위해 다양한 UI를 활용하는 것은 필요하나, 보다 중요한 것은 장애인 등 다양한 사용자가 동등하게 이용할 수 있도록 개발하거나 해당 기능을 대체할 수 있는 수단을 제공하는 등의 보편적 설계(Universal Design)를 적용해야 한다.

< Slide 기능에 대한 대체 수단 제공 사례 (두 번 두드리기(double tap)) >	< Slide 기능에 대한 대체 수단 미제공 사례 >
<div data-bbox="167 1498 510 1993"> </div> <div data-bbox="534 1563 782 1926"> <p>슬라이드 기능을 활용하지 못할 경우, VoiceOver를 이용할 경우에는 두 번 누르기(double tap)으로 해당 기능 활용 가능</p> </div>	<div data-bbox="810 1498 1117 1993"> </div> <div data-bbox="1141 1541 1412 1948"> <p>슬라이드 기능을 활용하지 못할 경우 대체 수단을 제공하지 않아, talkback을 이용하는 사용자의 경우에는 전화 받기 기능을 활용할 수 없음</p> </div>

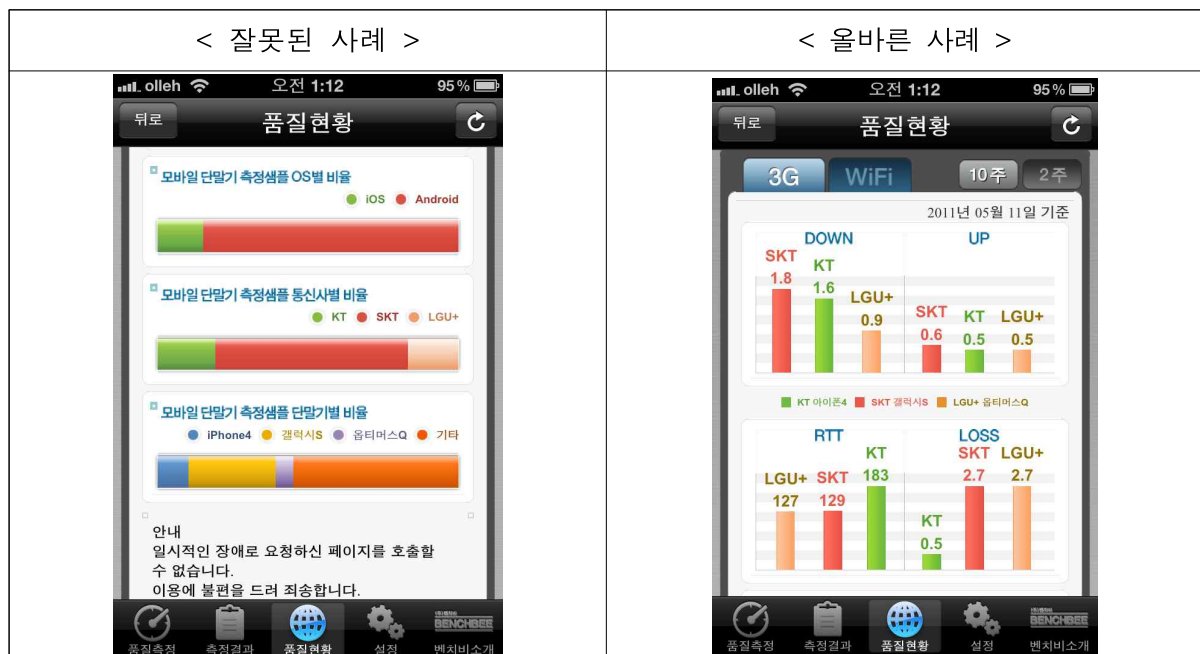
5. (색에 무관한 인식) 화면에 표시되는 모든 정보는 색에 관계없이 인식할 수 있어야 한다.

1) 준수 필요성 : 색맹 사용자의 경우 특정 색을 구별하지 못하는 경우가 있다. 그 대표적인 예로 적녹 색맹의 경우 적색과 녹색을 구분하지 못함으로서 일상 생활에서 신호등을 구분하지 못하고 이로 인하여 운전면허 취득도 어렵다. 이와 마찬가지로 모바일 애플리케이션에서 정보를 색으로만 제공할 경우 색맹이나 색약자의 경우 이를 인지할 수 없다.

2) 기술구현 방법 : 색상으로 정보를 구분할 경우, 색상 이외의 다른 방법으로도 동등한 내용을 전달할 수 있도록 설계한다. 색이외에 명암이나 텍스트, 특수기호 등을 색과 함께 제공하여 해당 정보를 인식할 수 있도록 제공해야 한다.

3) 구축 사례

오른쪽 아래의 그래프에서는 속도 값을 막대그래프로 표현하는데 막대그래프의 위에 의미를 명확히 알 수 있도록 통신사명과 막대그래프의 값을 텍스트로 제공하고 있어서 그 의미를 이해하는데 도움을 주고 있다. 색을 구분하기 어려운 사용자더라도 제공되는 텍스트 정보는 구분할 수 있기 때문에 그래프의 의미를 구분할 수 있다.



6. (명도 대비) 화면에 표시되는 모든 정보는 전경색과 배경색이 구분될 수 있도록 최소 대비 이상으로 제공되어야 한다.

1) 준수 필요성 : 시각장애인 등은 모바일 기기를 사용할 경우 화면에 표시되는 전경색과 배경색간의 구분이 잘 되지 않는 문제로 인하여 어려움을 겪는 경우가 많이 발생하고 있다. 특히 전경색과 배경색이 흰색과 회색, 노란색과 오렌지색 등으로 유사한 색으로 되어 있는 경우 이를 인지하기 매우 어렵다. 따라서 전경과 배경을 명확하게 인식할 수 있도록 콘텐츠를 제공해야 한다.

2) 기술구현 방법 : 명도 대비는 화면의 배경색과 객체를 표시하는 데에 사용되는 전경색 사이의 명도 차이의 비율(contrast)을 말한다. 모든 정보의 최소 대비는 3:1 이상이어야 한다. 저시력인에게 실효성을 가지기 위해서는 주 콘텐츠의 경우는 4.5:1 또는 7:1 이상이 되는 것이 바람직하다. 명도 대비를 높게 제공하기 어려운 경우에는 운영체제에서 제공하는 명도 대비를 활용할 수 있도록 제공하거나 애플리케이션 내에서 명도 대비를 높일 수 있는 설정 기능을 제공해야 한다.

※ iOS 4의 경우에는 “설정 → 일반 → 손쉬운 사용 → 검정색 바탕에 흰색”이라는 명도 대비 설정 기능을 제공하고 있다.

< 참고 : 색에 무관한 인식 관련 평가 도구 >

Color Doctor

<http://www.fujitsu.com/global/accessibility/assistance/cd/download.html>

Visual Impairment Simulator for Microsoft Windows

<http://vis.cita.uiuc.edu/index.php>

Colour Contrast Analyser

<http://juicystudio.com/article/colour-contrast-analyser-firefox-extension.php>

Colour Checker

<http://www.etre.com/tools/colourcheck/>

Color Selector

<http://www.fujitsu.com/global/accessibility/assistance/cs/download.html>

aDesigner

<http://www.alphaworks.ibm.com/tech/adesigner>


Accessibility Color Wheel

<http://gmazzocato.altervista.org/colorwheel/wheel.php>

3) 구축 사례

**은행 애플리케이션에서 인증서 가져오는 동작을 하면 아래의 왼쪽 그림과 같이 4자리의 숫자를 4번 입력하는 총 16자리의 인증번호가 나온다. 저 인증번호는 PC에서 휴대폰으로 인증서를 전송할 때 반드시 입력해야 하는데 모바일 화면에서 글씨/배경색의 대비가 충분하지 않기 때문에 저시력 시각장애인이 글씨를 읽는 것이 매우 힘들다. 이러한 경우 단순히 글씨를 알아보기 어렵다는 것에 그치는 것이 아니라 동작을 수행할 수 없는 결과를 낳는다. 저시력 사용자는 인증번호를 아예 알아보지 못하거나 잘못 읽어서 인증서를 휴대폰으로 전송하지 못하는 결과를 낳을 수 있다. 저시력 사용자에게 있어서 명도 대비가 없는 이러한 인증번호는 인증번호로서의 목적을 전혀 달성하지 못하고 있는 것이다.

** 모바일 고객센터에서 이용량을 보여주는 아래 오른쪽 그림을 살펴보면 그래프를 구분하는 왼쪽 라벨 부분에 문제가 있다. 라벨명인 사용일 수, 음성, 문자 메시지, 문자/멀티 메시지, MMS, 무선 인터넷 부분에서 배경색과 글자색의 대비가 충분하지 않아 저시력 사용자는 글씨를 읽기가 매우 어렵다. 폰트 크기도 작은데다 글자색과 배경색이 대비마저 낮아서 더 읽기가 어려운 상황이다. 라벨을 명확히 알아볼 수 없기 때문에 오른쪽에 나오는 그래프의 의미를 이해하는데 어려움을 겪을 수밖에 없다.

< 잘못된 사례 >	< 잘못된 사례 >
	

7. (자막, 수화 등의 제공) 멀티미디어 콘텐츠에는 동등한 내용의 자막, 원고 또는 수화가 제공되어야 한다.

1) 준수 필요성 : 청각장애인의 경우 동영상이나 음성으로 제공되는 콘텐츠에 대하여는 음성정보를 인지하기 어렵다. 따라서 이들에 대하여는 동기화된 자막을 제공하거나 별도의 원고를 제공하여야 콘텐츠를 인식할 수 있다. 멀티미디어 콘텐츠에 대한 동등한 내용의 자막, 원고 또는 수화는 시끄러운 환경이나 조용한 환경, 해당 언어를 모국어로 사용하지 않는 비장애인에게도 도움이 된다. 또한 동영상 검색시에도 자막이나 원고는 큰 도움을 줄 수 있다.

2) 기술구현 방법 : 멀티미디어 콘텐츠에는 동등한 내용의 자막, 원고 또는 수화를 제공해야 한다. 멀티미디어 콘텐츠를 요약한 자막이나 원고는 불충분하며, 멀티미디어에서 제공하는 음성 정보와 동등한 내용을 자막이나 원고로 제공해야 한다. 음성이 없이 동영상에 포함된 메시지나 자막, 중요 단어에 대한 강조, 홍보 문구 등은 시각장애인이 전혀 확인할 방법이 없으므로, 이러한 경우에는 반드시 화면해설 서비스(음성으로 메시지, 자막, 중요 단어 등을 제공)를 제공해야 한다. 자막, 수화 또는 원고는 화면 상의 콘텐츠와 동기화하여 제공하는 것이 바람직하다.

3) 구축 사례

왼쪽 아래의 그림은 ‘아이폰 용 복지 TV 애플리케이션’의 사례이다. 본 애플리케이션에서 제공하는 동영상의 경우 자막과 함께 수화를 동시에 제공하여 장애인의 접근성을 확보한 우수 사례이다.

< 올바른 사례(복지 TV 애플리케이션) >	< 잘못된 사례 (자막 미제공) >
	

나. 권고사항(8개) : 모바일 애플리케이션의 접근성 향상을 위하여 무리한 부담이 되지 않는 한 준수할 것을 권장하는 사항을 말한다.

1. (Native UI Component) 운영체제에서 제공하는 기본 사용자 인터페이스 컴포넌트(Native UI Component)를 최대한 이용하는 것이 바람직하다.

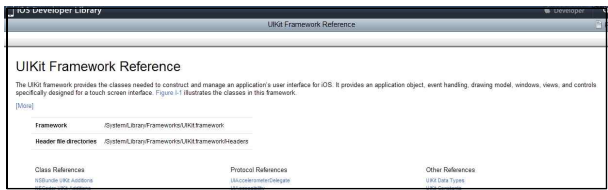
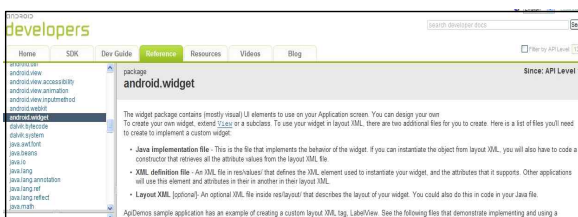
1) 준수 필요성 : 대부분의 운영체제에서 제공하는 기본 사용자 인터페이스 컴포넌트(Native UI Component)는 접근성을 고려하여 제공되고 있다. 애플리케이션 개발 시 비용과 시간 등의 제약이 있는 경우 운영체제에서 제공하는 기본 사용자 인터페이스 컴포넌트를 잘 활용하는 것이 좋다. 이에 반해 사용자 변형 UI 컴포넌트(Custom UI Component)를 사용하는 경우에는 운영체제에서 제공하는 VoiceOver 등이 동작하지 않을 수 있다. 그러므로 최대한 기본 사용자 인터페이스를 활용하는 것이 바람직하다.

2) 기술구현 방법 : 모바일 애플리케이션을 개발함에 있어 기본 사용자 인터페이스 컴포넌트를 최대한 활용하는 것이 접근성을 높이는데 도움이 된다.

① 애플의 iOS에서 제공 방법 : Native UI Component에는 UIWindow, UILabel, UIPickerView 등이 있다. 특히 웹 페이지를 내장하는 페이지를 만들 경우에는 UIWebView를 통해 작성을 하게 된다.

② 구글의 안드로이드에서 제공 방법 : Native UI Component에는 View, ImageView 등이 있다.

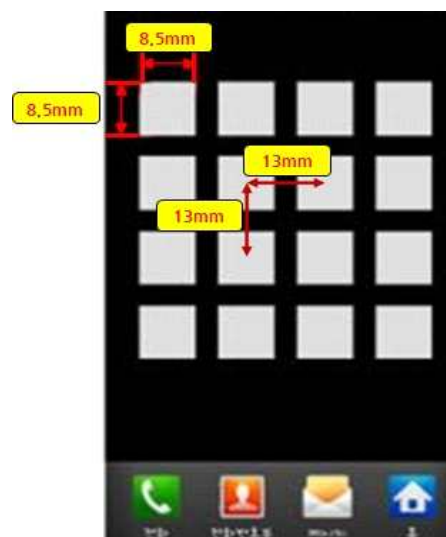
3) 구축 사례 : 대표적인 애플과 구글의 운영체제에서 제공하는 Native UI Component는 아래의 그림과 같다.

< iOS에서의 Native UI Component >	< 안드로이드에서의 Native UI Component >
 <p>http://developer.apple.com/library/ios/#documentation/UIKit/Reference/UIKit_Framework/_index.html#//apple_ref/doc/uid/TP40006955</p>	 <p>http://developer.android.com/reference/android/widget/package-summary.html</p>

2. (컨트롤간 충분한 간격) 컨트롤은 충분한 간격으로 배치하는 것이 바람직하다.

1) 준수 필요성 : 모바일 기기의 경우 터치를 통하여 컨트롤하기 때문에 터치의 정확성이 매우 중요하다. 터치의 정확성을 담보하기 위해서는 좌표 값도 중요하지만 각 컨트롤간의 간격도 중요하다. 비장애인도 각 버튼간의 터치 간격이 좁게 되어 있고 버튼의 크기가 작으면 매우 불편한데 뇌성마비 장애인의 경우 손 떨림 등이 많이 있고 이로 인하여 정확한 터치 동작을 취하기가 매우 어렵다. 따라서 터치 동작을 원활히 수행할 수 있도록 터치 간격을 적절히 유지하는 것이 매우 중요하다. 저시력인의 경우도 터치 타겟이 작을 경우 많은 어려움을 겪게 되는데 특히 중요한 결제 정보를 입력하거나 은행용 모바일 애플리케이션을 사용하는 경우 실수로 잘못 터치하는 경우 되돌이킬 수 없는 문제가 발생할 수 도 있다.

2) 기술구현 방법 : 좁은 화면 공간의 경우, 사용자의 의도와 무관하게 다른 컨트롤을 누르게 되는 문제가 발생할 수 있으므로, 이를 피하기 위해서 컨트롤 사이의 공간을 충분히 확보하여 사용자가 컨트롤 영역을 명확히 구분할 수 있도록 하는 것이 바람직하다. 컨트롤 중심간 간격은 13mm X 13mm 이상을 권장한다. 선택해야 하는 컨트롤 영역의 크기는 8.5mm X 8.5mm 이상을 권장한다. 컨트롤의 터치 공간을 충분히 확보하여 사용자가 컨트롤 영역을 명확히 구분할 수 있도록 제공하는 것이 좋다. 사용자의 의도에 따라 컨트롤을 확대하여 사용할 경우 상대적인 크기로 커져서 손쉽게 활용할 수 있도록 제공하는 것이 좋다. 쿼티 입력 등 운영체제에서 제공하는 기본 사용자 인터페이스의 경우에는 예외로 한다.



3. (알림 기능) 사용자에게 알림을 제공할 때에는 진동, 시각, 소리 등 최대한 다양한 방법으로 사용자가 선택할 수 있도록 제공하는 것이 바람직하다.

1) 준수 필요성 : 한 가지 감각으로만 정보를 제공하는 경우에는 다양한 사용자가 이를 활용할 수 있다고 보장하기 어렵다. 예를 들어 시력을 요하는 정보를 제공하거나 음성으로만 정보를 제공하는 경우 시각장애인이나 청각장애인들은 이러한 정보를 인지할 수 없다. 따라서 알림 정보 등을 제공하는 경우에는 소리나 화면 진동 등 다양한 방법을 동시에 사용하여 정보를 제공하는 것이 필요하다.

2) 기술구현 방법 : 사용자에게 알림을 제공할 때에는 한 가지 감각에만 의존하지 말고 다양한 감각이나 표현 방법을 통해 사용자가 원하는 알림 기능을 제공하는 것이 바람직하다. 사용자에게 다양한 방법으로 알림을 제공될 수 있도록 시각, 청각, 촉각 등의 피드백을 제공해야 하며, 다양한 알림 기능을 제공할 경우 적절한 방법을 사용자가 선택할 수 있도록 제공하는 것이 더 바람직하다.

3) 구축 사례

페이스북 애플리케이션에서는 알림이 상당히 세분화 되어있다. 사용자가 자신에게 적합한 방법으로 알림 정보를 진동, 소리 등으로 정보를 얻을 수 있도록 기능을 제공하고 있다. 또한 이러한 알림 정보는 가급적 운영체제에서 제공하는 Native UI를 활용하는 것이 바람직하다.



4. (범용 폰트 이용) 폰트의 크기 조절, 확대 기능을 제공하거나 운영체제에서 제공하는 관련 기능을 활용할 수 있는 방법을 제공하는 것이 바람직하다.

1) 준수 필요성 : 저시력인, 고령자 등은 일반 PC보다 화면이 적은 모바일 기기를 활용함에 있어 작은 화면으로 인해 정보를 인식하는데 어려움을 겪는다. 저시력인이나 고령자 등도 동등하게 모바일 애플리케이션을 이용할 수 있기 위해서는 폰트를 바꾸거나 폰트의 크기 등을 조절하여 자신에게 적합한 방법으로 애플리케이션을 이용할 수 있어야 한다.

2) 기술구현 방법 : 절대 폰트를 사용하지 말고, 사용자 선택에 따라 폰트의 크기를 변화시킬 수 있도록 제공하는 것이 바람직하다. 시스템이나 사용자가 선택한 환경(Setting)을 그대로 상속(Inherit)할 수 있도록 제공해야 한다. 범용 폰트(Global Font)는 운영체제에 내장되어 확대나 축소, 기울임 등의 변형 형태가 제공되는 글자체를 말한다. 모든 애플리케이션 화면에서 폰트 크기의 조절이 가능하도록 설계하거나, 최소한 확대 기능을 제공한다. 폰트 크기 조절을 용이하게 하기 위해서는 텍스트 이미지보다 폰트가 지정되어 있는 텍스트를 사용하는 것이 바람직하다. 폰트의 확대 시 텍스트의 내용이나 기능의 손실 없이 최소 200%까지 확대될 수 있도록 제공하는 것이 좋다. 또한 별도의 폰트를 사용하는 경우 사용자가 인지하기 어렵게 되는 경우가 발생할 수 있으니 운영체제에서 제공하는 글로벌 폰트를 이용하는 것이 바람직하다.

① 애플의 iOS에서 제공되는 글로벌 폰트 종류('11년 7월말 기준)


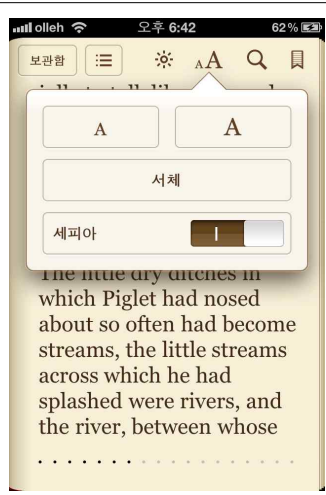
Font Family: American Typewriter Font:AmericanTypewriter Font:AmericanTypewriter-Bold Font Family: AppleGothic Font:AppleGothic Font Family: Arial Font:ArialMT Font:Arial-BoldMT Font:Arial-BoldItalicMT Font:Arial-ItalicMT Font Family: Arial Rounded MT Bold Font:ArialRoundedMTBold Font Family: Arial Unicode MS Font:ArialUnicodeMS Font Family: Courier Font:Courier Font:Courier-BoldOblique Font:Courier-Oblique Font:Courier-Bold Font Family: Courier New Font:CourierNewPS-BoldMT Font:CourierNewPS-ItalicMT Font:CourierNewPS-BoldItalicMT	Font:HiraKakuProN-W6 Font Family: Marker Felt Font:MarkerFelt-Thin Font Family: STHeiti J Font:STHeitiJ-Medium Font:STHeitiJ-Light Font Family: STHeiti K Font:STHeitiK-Medium Font:STHeitiK-Light Font Family: STHeiti SC Font:STHeitiSC-Medium Font:STHeitiSC-Light Font Family: STHeiti TC Font:STHeitiTC-Light Font:STHeitiTC-Medium Font Family: Times New Roman Font:TimesNewRomanPSMT Font:TimesNewRomanPS-BoldMT Font:TimesNewRomanPS-BoldItalicMT Font:TimesNewRomanPS-ItalicMT Font Family: Trebuchet MS Font:TrebuchetMS-Italic Font:TrebuchetMS
--	---

Font:CourierNewPSMT Font Family: DB LCD Temp Font:DBLCDTempBlack Font Family: Georgia Font:Georgia-Bold Font:Georgia Font:Georgia-BoldItalic Font:Georgia-Italic Font Family: Helvetica Font:Helvetica-Oblique Font:Helvetica-BoldOblique Font:Helvetica Font:Helvetica-Bold Font Family: Helvetica Neue Font:HelveticaNeue Font:HelveticaNeue-Bold Font Family: Hiragino Kaku Gothic **** W3 Font:HiraKakuProN-W3 Font Family: Hiragino Kaku Gothic **** W6	Font:Trebuchet-BoldItalic Font:TrebuchetMS-Bold Font Family: Verdana Font:Verdana-Bold Font:Verdana-BoldItalic Font:Verdana Font:Verdana-Italic Font Family: Zapfino Font:Zapfino
--	---

② 구글의 안드로이드에서 제공되는 글로벌 폰트 종류('11년 7월말 기준)

"sans-serif" "arial" "helvetica" "tahoma" "verdana" "serif" "times" "times new roman" "palatino" "georgia"	"baskerville" "goudy" "fantasy" "cursive" "ITC Stone Serif" "monospace" "courier" "courier new" "monaco"
---	--

3) 구축 사례 : 트위터, iBooks 애플리케이션에서는 저시력자 등을 위해 폰트의 크기 조절 기능을 설정할 수 있도록 제공하고 있다.

< 트위터의 글자크기 설정 화면 >	< iBooks 폰트 크기 및 서체 설정 화면 >
	

5. (사용자 인터페이스의 일관성) 사용자 인터페이스 요소들의 배치를 일관성 있게 제공하는 것이 바람직하다.

1) 준수 필요성 : 저시력인이나 고령자 등 화면을 확대하는 사용자의 경우에는 전체 화면이 아니라 창의 일부 영역을 화면에 확대하여 이용한다. 따라서 애플리케이션 창마다 내비게이션 컨트롤의 위치와 모양이 다르다면 새로운 애플리케이션 창으로 이동할 때마다 사용법을 익히는 데 많은 어려움이 있을 것이다. 또한 지적 장애인의 경우 방문한 웹 페이지별로 메뉴와 내비게이션 컨트롤의 위치나 모양이 바뀌게 되면 사용자는 이들 웹 페이지를 동일한 웹 사이트가 제공하는 웹 페이지로 인식하기보다는 새로운 웹 사이트가 제공하는 웹 페이지로 인식할 가능성이 높다. 이에 일관성 있는 사용자 인터페이스를 제공하는 것이 바람직하다.

2) 기술구현 방법 : 사용자 경험(User Experience)에 비추어 일관성 있는 사용자 인터페이스(UI) 요소를 제공하는 것이 바람직하다. 사용자 인터페이스를 구성하고 있는 요소들(폰트, 크기, 화면 색상, 링크 제공 방법, 이모티콘 등)을 사용자가 다시 학습하지 않아도 되도록 해당 애플리케이션 내에서 일관성 있게 제공하는 것이 바람직하다.

3) 구축 사례

*** 영화관 모바일 애플리케이션으로 예약 시간을 선택하는 선택 창과 인원 및 좌석을 선택하는 창이 일관성 없이 제공되어 있어 사용자에게 혼란을 주는 사례이다.

< 일관성 없는 입력 서식 제공 사례 (iOS Native UI Component (좌측), Custom UI Component (우측)) >



6. (깜빡거림의 사용 제한) 광과민성 발작을 일으킬 수 있는 콘텐츠를 제공하지 않는 것이 바람직하다.

1) 준수 필요성 : 깜빡이거나 번쩍이는 콘텐츠를 제공할 경우 특정 사용자에게 광과민성 발작을 일으킬 우려가 있다. 예를 들어 일본에서 방영된 만화 영화인 포켓몬에서 깜빡임과 번쩍임이 과도하게 제공되어 아동들이 병원에 가서 치료를 받았던 사례가 있다. 그러므로 깜빡임과 번쩍임이는 효과로 정보를 제공하기 보다는 효과적인 디자인 등을 통해 모바일 애플리케이션의 정보를 제공하는 것이 바람직하다.

2) 기술구현 방법 : 깜빡이거나 번쩍이는 콘텐츠를 제공해야만 할 경우 초당 3-50회 주기는 피해서 제공하는 것이 좋다. 깜빡이거나 번쩍이는 콘텐츠를 사용해야 할 경우, 사전에 경고를 하고 깜빡임이나 번쩍임을 회피할 수 있는 수단을 제공하는 것이 바람직하다. 이러한 콘텐츠는 깜빡이는 배경이나 텍스트, 꺼지고 켜짐을 반복적으로 수행하는 그래픽, 또는 다른 여러 이미지를 반복적으로 보여주는 모든 것들을 포함한다.

7. (배경음 사용금지) 자동으로 재생되는 배경음을 사용하지 않는 것이 바람직하다.

1) 준수 필요성 : 시각장애인의 경우 모바일 기기나 모바일 화면낭독 프로그램에서 제공하는 음성읽기 기능(iOS VoiceOver, 안드로이드 Talkback, 심비안 및 윈도우 모바일 6.5에서 활용되는 Code Factory사의 Mobile Speak 등)을 활용하여 모바일 애플리케이션을 이용한다. 이러한 기능들은 기본적으로 음성을 통하여 정보를 제공하고 있다. 이에 따라 애플리케이션을 실행하였을 때 배경음이 자동적으로 나오게 되면 음성으로 정보를 정확히 전달받을 수 없다. 따라서 이러한 배경음이 필요한 경우 이에 대한 알림 정보를 제공하고, 사용자가 선택할 경우에만 실행이 되도록 제공하는 것이 바람직하다.

2) 기술구현 방법 : 자동으로 재생되는 배경음(동영상, 음성, 음악 등)을 사용하지 않는 것이 바람직하다. 단, 3초 미만의 배경음은 예외로 한다. 배경음을 사용할 경우, 반드시 배경음을 제어할 수 있는 수단(멈춤, 일시정지, 음량조절 등)이나 배경음 제어로 이동하는 링크를 애플리케이션 첫 부분에 제공하는 것이 좋다. 또한 음량 조절은 운영체제에서 제공하는 음량과 독립적으로 배경음만 조절할 수 있도록 제공하는 것이 더 좋다.

3) 구축 사례

** 자동차 회사에서 제공하는 모바일 애플리케이션으로 시작화면부터 사용자가 선택하지 않았음에도 불구하고 배경음이 자동적으로 흘러나오고 있다. 이를 해지하기 위해서는 설정 메뉴로 들어가서 음악과 효과음을 줄여야만 정지되는 형태로 시각 사용자들에게 혼란을 주는 사례이다.

< 자동 배경음을 사용한 잘못된 사례 >	< 해결 방안 >
	<p>동영상을 사용자의 선택에 의해 활성화되도록 제공하는 것이 바람직하다.</p> <p>배경음을 제공할 경우에는 반드시 애플리케이션 첫 부문에 이를 정지할 수 있는 기능을 제공하는 것이 바람직하다.</p>

8. (장애인 사용자 평가) 애플리케이션 개발 시 다양한 모바일 기기에서의 이용 가능 여부를 점검해야 하며, 장애인 테스트를 수행하는 것이 바람직하다.

1) 준수 필요성 : 모바일 애플리케이션을 개발함에 있어 앞에서 말한 모든 지침 항목을 준수하여 개발하였다 하더라도 장애인 사용자가 실제로 사용할 수 있는지 점검하는 것이 중요하다. 이는 장애인이 가지고 있는 특성에 대한 이해의 차이 때문에 실제로는 접근성 지침을 지켜도 장애인이 사용하기 어려울 수 있기 때문이다. 또한 사용자는 모바일 애플리케이션을 다양한 모바일 기기를 활용하여 접근할 것이다. 이에 가급적 많은 모바일 기기를 활용하여 애플리케이션의 정보 접근 및 기능 가능여부를 사전에 점검하는 작업을 하는 것이 필요하다.

2) 준수 방법 : 기획 단계부터 해당 애플리케이션에 대해 장애인 사용자 평가를 수행하는 것이 바람직하다. 시각, 청각, 상지 장애 등의 다양한 장애인 사용자를 대상으로 하는 것이 좋으며, 고령자도 포함하는 것이 좋다. 또한 모바일 애플리케이션 개발자, 운영자 등도 운영체제에서 제공하는 애플의 VoiceOver, 구글의 TalkBalk 등을 활용하여 지속적으로 점검하는 것이 바람직하다.