

N-Screen 고품질 브라우징 기술

- 이정표(케이티하이텔 플랫폼사업부)
- 김민태(케이티하이텔 iOS팀 및 Web플랫폼Lab)
- 이원주(인하공업전문대학 컴퓨터정보과)

I. N-Screen 서비스 개요

N-스크린 고품질 크로스 브라우징의 기술 설계 내용으로 크게 아래와 같이 3가지 부분을 다룬다.

- 브라우저 호환성 유지 방법
- SVG를 Canvas로 변환하는 방법
- 그래픽요소 벤치마크

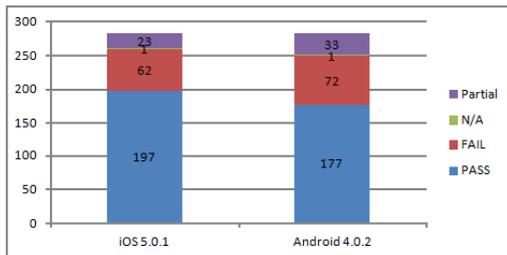


그림 1. HTML5 지원현황

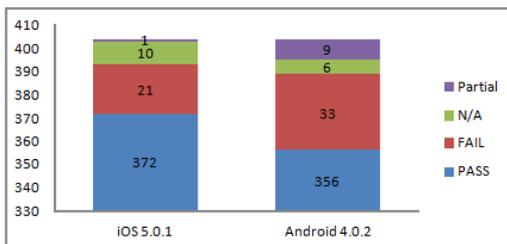


그림 2. CSS3 지원현황

그림 1과 2에서 보는 바와 같이 대표적인 모바일 플랫폼인 iOS와 Android의 최신 브라우저에서도 HTML5 및 CSS3 웹 표준이 정상적으로 지원되지 않고 있다. 이러한 브라우저 간의 비호환성으로 인해 N-스크린 콘텐츠 개발에 어려움이 많다. N-스크린 콘텐츠 개발을 위하여 그림 3에 보는 바와 같이 크로스 브라우징을 위한 호환성 유지 방법에 대한 개발자 가이드까지 나오고 있는 상황이다[1]. 이러한 브라우저 호환성 유지가 어려운 이유 때문에 별도의 호환성 라이브러리를 통해 N-스크린 콘텐츠의 호환성을 확보해야 한다.

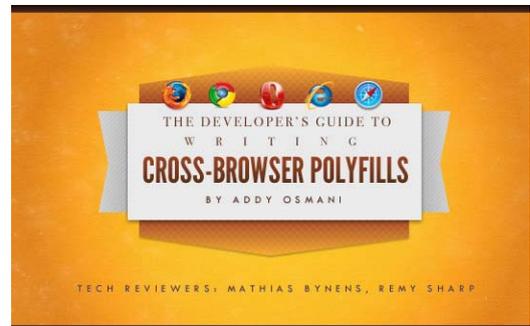


그림 3. 호환성 가이드 웹 페이지

플랫폼간의 단편화 현상을 해결하기 위한 방안으로 HTML5가 화두가 되었고, 그것을 가능하게 하고자 여러 브라우저 제조사가 개발 중에 있다. 하지만, 아직까지는 호환성 문제를 해결하지 못하고 있기 때문에 N-스크린 콘텐츠를 사용하여 여러 단말 환경에서도 호환성을 유지할 수 있는 방법

을 찾고자 한다. 이러한 방법으로 개발자는 브라우저간의 비 호환성 문제를 해결하고, 콘텐츠 유통 및 배포는 호환성 유지를 통한 유지보수 비용을 절감할 수 있다.

1. Canvas로 변환

현재 브라우저 중에서 SVG(Scalable Vector Graphics)[2][3]를 지원하지 않는 브라우저가 많다. 특히, 사용자 층이 많은 Android 2.3 플랫폼에서도 SVG를 지원하지 않는다. 아래 그림은 SVG를 지원하고 있는 브라우저에 설명이다.



그림 4. SVG 지원현황

SVG를 지원하지 않는 단말일지라도, HTML5 Canvas가 지원된다면, SVG를 Canvas로 변경하여 콘텐츠 이용이 가능하다. 즉, SVG 콘텐츠를 보다 폭 넓게 이용할 수 있다.

SVG는 웹표준 규격으로 TV등의 큰 화면에서는 매우 효과적으로 사용될 수 있다. 하지만 브라우저 지원 제약으로 인해서 해당 콘텐츠를 재생할 수 없는 경우에는 콘텐츠 유통의 큰 제약 사항이 될 것이다. 따라서 SVG를 Canvas로 변환하는 기술을 도입하여, SVG가 지원되지 않는 단말에서도 Canvas를 활용한 콘텐츠 재생이 가능하도록 함으로써 콘텐츠 유통에 제약 사항을 제거해야 한다.

2. 그래픽요소 벤치마크

여러 호환성 요소 가운데, 실제 미디어 콘텐츠에서 가장 중요한 부분은 그래픽 요소이다. Canvas 같은 요소의 경우 여러 단말 간의 성능 차이 때문에 호환성 유지에 있어 문제가 생길 수 있다.

그래픽 요소 벤치마크를 통하여, 다양한 그래픽요소에 대한 성능을 확인함으로써 N-스크린 콘텐츠 제작 및 호환성 설계 등에 활용한다. 또한, 그래픽 요소(이미지, DIV, Canvas

등)가 포함된 콘텐츠 제작 및 배포 과정에 가이드라인을 제공하여, 미리 성능 및 품질을 검증함으로써 콘텐츠의 신뢰성을 높일 수 있다.

II. 브라우저 호환성 유지 기술

1. 브라우저 호환성 유지 필수 항목

브라우저 호환성 유지 방법 설계를 위해서는 먼저 브라우저 간의 호환성이 얼마나 유지되고 있는지에 대한 분석이 필요하다. 대표적인 N-스크린 단말로 Android 및 iOS 플랫폼 기반의 폰, 태블릿의 브라우저 표준 규격 준수 여부를 확인하여, 호환성에 문제가 되는 부분을 분석한 결과 약 20% 정도의 요소들에 대해서 호환성 문제가 발생한다[4].

하지만, Android 폰에서 지원되지만, iOS 폰에서는 지원되지 않는다면, iOS 태블릿(iPad)에서는 지원 하지만 Android 최신 버전인 ICS(Icecream Sandwich, Android 4.0)에서는 지원하지 않는 항목에 대해서는 호환성을 위한 별도의 기술이 필요하다.

1.1 HTML5 Audio

audio 요소는 소리 또는 오디오 스트림을 나타내며, audio 요소 내부에 콘텐츠를 제공할 수 있다. 이렇게 하면 audio 요소를 지원하지 않는 구형 버전의 브라우저들은 오디오 플레이어들을 시도해 보거나 사용자가 오디오 콘텐츠에 직접 접근하는 방법을 텍스트로 제공할 수 있다.

```
[NamedConstructor=Audio()],
NamedConstructor=Audio(in DOMString src)
interface HTMLAudioElement : HTMLMediaElement {};
```

그림 5. HTML5 Audio

1.2 HTML5 Button

button 요소는 버튼을 나타낸다. 기본값은 제출 버튼 상태이다. type 속성이 제출 버튼 상태이면, 요소는 제출 버튼이다.

```

DOM interface:
interface HTMLButtonElement : HTMLElement {
    attribute boolean autofocus;
    attribute boolean disabled;
    readonly attribute HTMLFormElement form;
    attribute DOMString formAction;
    attribute DOMString formEnctype;
    attribute DOMString formMethod;
    attribute DOMString formNoValidate;
    attribute DOMString formTarget;
    attribute DOMString name;
    attribute DOMString type;
    attribute DOMString value;

    readonly attribute boolean willValidate;
    readonly attribute ValidityState validity;
    readonly attribute DOMString validationMessage;
    boolean checkValidity();
    void setCustomValidity(in DOMString error);

    readonly attribute NodeList labels;
};
interface HTMLAudioElement : HTMLMediaElement {};

```

그림 6. HTML5 Button

1.3 HTML5 datalist

datalist 요소는 다른 컨트롤에서 사용할 수 있도록 미리 정의된 옵션 집합이다. 요소의 내용은 미리 정의된 옵션을 뜻하는 **option** 요소와 섞여서 구형 브라우저를 위한 폴백 요소를 나타낸다. 렌더링 과정에서는 아무것도 나타내지 않는다.

```

interface HTMLDataListElement : HTMLElement {
    readonly attribute HTMLCollection options;
};

```

그림 7. HTML5 datalist

1.4 HTML5 form

form 요소는 폼과 관련된 요소들의 집합이다. 그중 일부는 서버에 전송하여 처리할 변경 가능한 값이다. 폼은 텍스트 필드, 버튼, 체크박스, 범위 컨트롤, 색상 선택기 같은 폼 컨트롤들을 갖고 있는 웹 페이지의 컴포넌트이다. 사용자는 폼과

상호작용하고, 서버에서 처리될 데이터를 제공할 수 있다. 검색 결과 혹은 계산값을 돌려주는 것을 예로 들 수 있다. 대개의 경우 클라이언트 사이드 스크립트는 필요가 없지만, 사용자 경험을 증진시키거나, 또는 데이터를 서버로 보내지 않을 목적으로 사용할 수 있는 API들이 있다.

```

[OverrideBuiltins]
interface HTMLFormElement : HTMLElement {
    attribute DOMString acceptCharset;
    attribute DOMString action;
    attribute DOMString autoComplete;
    attribute DOMString enctype;
    attribute DOMString encoding;
    attribute DOMString method;
    attribute DOMString name;
    attribute boolean noValidate;
    attribute DOMString target;

    readonly attribute HTMLFormControlsCollection elements;
    readonly attribute long length;
    caller getter any (in unsigned long index);
    caller getter any (in DOMString name);

    void submit();
    void reset();
    boolean checkValidity();

    void dispatchFormInput();
    void dispatchFormChange();
};

```

그림 8. HTML5 form

1.5 HTML5 menu

menu 요소는 커맨드 목록이다. **type** 속성은 나열 속성이며 선언되는 메뉴의 종류를 나타낸다. 속성은 3개의 상태, 즉 **context**, **toolbar**, **list**의 상태 속성이 있다. 각각의 상태는 동일한 이름의 키워드와 연관관계가 있다. 속성을 생략하였을 경우의 기본값은 **list** 상태이다. 이것은 단순히 커맨드의 목록일 뿐이며, 컨텍스트 메뉴를 선언하거나 툴바를 정의하지는 않는다. **label** 속성은 메뉴에 레이블을 부여한다. 사용자 에이전트는 UI에서 중첩된 메뉴의 레이블에 이것을 사용한다. 예를 들어 다른 메뉴를 포함하는 컨텍스트 메뉴는 중첩된 메뉴의 **label** 속성을 이용하여 서브메뉴의 메뉴 레이블을 나타낸다.

```
interface HTMLMenuElement : HTMLElement {
    attribute DOMString type;
    attribute DOMString label;
};
```

그림 9. HTML5 menu

1.6 HTML5 ol

ol 요소는 의도적으로 순서를 갖게끔 만든 목록을 나타낸다. 이러한 순서를 바꾸면 문서의 의미도 바뀐다. ol 요소의 자식 노드인 li 요소가 트리 순서로 목록을 구성할 수 있도록 한다.

```
interface HTMLListElement : HTMLElement {
    attribute boolean reversed;
    attribute long start;
    attribute DOMString type;
};
```

그림 10. HTML5 ol

1.7 HTML5 progress

progress 요소는 작업의 진척도를 나타낸다.

```
interface HTMLProgressElement : HTMLElement {
    attribute double value;
    attribute double max;
    readonly attribute double position;
    readonly attribute HTMLFormElement form;
    readonly attribute NodeList labels;
};
```

그림 11. HTML5 progress

이 진척도는 아래와 같이 두 가지로 구분된다.

- ① 작업이 수행되고 있지만 작업이 완료되기 전에는 얼마나 많은 일이 남아있는지 알 수 없는 모호한 경우(예를 들어 원격 서버의 응답을 기다려야 한다.)
- ② 진척도가 0 부터 최대값 사이의 숫자로서 표현될 수 있는 경우.

요소에는 현재 작업의 진척도를 표현할 수 있는 두개의 속성이 있다. value 속성은 작업이 얼마나 진척되었는지를 나타낸다. max 속성은 전체적으로 얼마나 많은 작업이 필요한

지 나타낸다. 단위는 임의의 것이며 명시되지 않는다. 현재의 값과 최대값을 요소 내부에 인라인으로도 포함시켜, 구형 브라우저의 사용자들도 진척도를 알 수 있다.

1.8 HTML5 rt

rt 요소는 ruby 텍스트의 ruby 주석이다. rt 요소는 ruby 요소 내에서 자신의 바로 앞에 나타나는 0 이상의 노드에 대한 주석(자신의 자식 요소로 주어지는)을, rp 요소를 무시하면서 나타낸다.

Uses HTMLElement.

그림 12. HTML5 rt

1.9 HTML5 video

video 요소는 비디오, 혹은 영화를 재생하기 위해 사용된다. video 요소를 지원하지 않는 오래된 웹 브라우저에서 구형 비디오 플러그인을 시도하거나, 사용자가 비디오 콘텐츠에 접근할 수 있는 방법을 제시하는 텍스트를 보여줄 수 있도록 video 요소 내부에 콘텐츠를 제공할 수 있다.

```
interface HTMLVideoElement : HTMLMediaElement {
    attribute unsigned long width;
    attribute unsigned long height;
    readonly attribute unsigned long videoWidth;
    readonly attribute unsigned long videoHeight;
    attribute DOMString poster;
    [PutForwards=value attribute DOMSettableTokenList audio;
};
```

그림 13. HTML5 video

1.10 HTML5 history

문맥에 포함된 Document 들의 연속을 세션 히스토리라고 한다. History 객체는 문맥의 세션 히스토리에 들어 있는 페이지들을 나타낸다. 각각의 문맥(중첩된 문맥을 포함)은 유일한 세션 히스토리를 갖는다. 문맥의 세션 히스토리에 들어 있는 각각의 Document 객체는 History 객체의 유일한 인스턴트와 연결된다. 이러한 것들은 모두 동일한 세션 히스토리 모델을 따라야 한다. History 객체는 자신의 문맥의 세션 히스토리를 세션 히스토리 항목의 평면화된 목록으로 나타낸다. 각

각의 세션 히스토리 항목은 URL, 상태 객체, 또는 둘 모두로 구성된다.

```
interface History {
  readonly attribute long length;
  void go(in optional long delta);
  void back();
  void forward();
  void pushState(in any data, in DOMString title, in
    optional DOMString url);
  void replaceState(in any data, in DOMString title, in
    optional DOMString url);
};
```

그림 14. HTML5 history

1.11 HTML5 file

웹 애플리케이션이 로컬 파일에 접근할 수 있도록 한다. 파일을 업로드하기 위해 웹 사이트의 특정 영역으로 파일을 드래그 하거나 <input> 요소로부터 전달받은 파일에 접근하여 파일의 이름, 경로, 크기, 종류 등에 대한 정보를 취득할 수 있다. 물론, 읽기전용 상태로 접근된 것이기 때문에 실제 파일의 물리적 변형은 일어나지 않는다.

```
interface FileList {
  getter File? item(unsigned long index);
  readonly attribute unsigned long length;
};
[Constructor]
interface FileReader: EventTarget {
  // async read methods
  void readAsArrayBuffer(Blob blob);
  void readAsBinaryString(Blob blob);
  void readAsText(Blob blob, optional DOMString
    encoding);
  void readAsDataURL(Blob blob);
  void abort();
  // states
  const unsigned short EMPTY = 0;
  const unsigned short LOADING = 1;
  const unsigned short DONE = 2;
  readonly attribute unsigned short readyState;
  // File or Blob data
  readonly attribute any result;
  readonly attribute DOMError error;
  // event handler attributes
  attribute [TreatNonCallableAsNull] Function? onloadstart;
  attribute [TreatNonCallableAsNull] Function? onprogress;
  attribute [TreatNonCallableAsNull] Function? onload;
  attribute [TreatNonCallableAsNull] Function? onabort;
  attribute [TreatNonCallableAsNull] Function? onerror;
  attribute [TreatNonCallableAsNull] Function? onloadend;
};
```

그림 15. HTML5 file

1.12 HTML5 WebSocket

웹 소켓은 웹 서버와 웹 브라우저가 지속적으로 연결된 TCP 라인을 통해 실시간으로 데이터를 주고받을 수 있도록 하는 HTML5의 새로운 사양이다. 따라서 웹 소켓을 이용하면 일반적인 TCP소켓과 같이 연결지향 양방향 전이중 통신이 가능하다.

```
[Constructor(in DOMString url, in optional DOMString
  protocols)]
[Constructor(in DOMString url, in optional DOMString[]
  protocols)]
interface WebSocket {
  readonly attribute DOMString url;

  // ready state
  const unsigned short CONNECTING = 0;
  const unsigned short OPEN = 1;
  const unsigned short CLOSING = 2;
  const unsigned short CLOSED = 3;
  readonly attribute unsigned short readyState;
  readonly attribute unsigned long bufferedAmount;

  // networking
  attribute Function onopen;
  attribute Function onmessage;
  attribute Function onerror;
  attribute Function onclose;
  readonly attribute DOMString protocol;
  void send(in DOMString data);
  void close();
};
WebSocket implements EventTarget;
```

그림 16. HTML5 WebSocket

1.13 HTML5 Page Visibility

사용자가 브라우저를 최소화하거나 다른 탭으로 이동해서 페이지가 보이지 않을 때 처리가 가능하다. 즉, 페이지의 상태가 'hidden', 'visible', 'preview', 'prerender' 인지 확인을 할 수 있는 기능이다.

```
[NoInterfaceObject]
interface DocumentVisibility {
  const DOMString PAGE_HIDDEN = "hidden";
  const DOMString PAGE_VISIBLE = "visible";
  const DOMString PAGE_PREVIEW = "preview";
  const DOMString PAGE_PRERENDER = "prerender";

  readonly attribute boolean hidden;
  readonly attribute unsigned short visibilityState;
};
Document implements DocumentVisibility;
```

그림 17. HTML5 Page Visibility

1.14 CSS3 background

- background-color
 - 요소의 배경색을 설정하는 속성
- background-image
 - 요소의 배경 이미지를 설정하는 속성
- background-repeat
 - 배경 이미지의 반복을 설정하는 속성
- background-attachment
 - 배경을 어떤 식으로 붙일지 설정하는 속성
- background-position
 - 배경 이미지의 최초 위치를 지정하는 속성
- background-clip
 - 배경 그리는 영역을 설정하는 속성
- background-origin
 - 각 요소들에 대한 배경 위치를 설정하는 속성
- background-size
 - 배경 이미지의 크기를 설정하는 속성

1.15 CSS3 Animation

웹 페이지에 애니메이션을 구현할 수 있는 특성이다. 키프레임으로 동적효과를 위한 각 변화의 지점을 정의한 후 이 프레임들이 어떤 식으로 연결되는지를 설정하는 애니메이션 속성을 정의해야 한다. 애니메이션 속성으로 애니메이션이 실행되는 시간, 반복 횟수, 변화의 정도, 프레임 연결 방향등을 지정하게 된다.

- webkit-animation-name
 - 키 프레임을 지정한다. 앞서 키 프레임의 이름을 여기에 매핑한다.
- webkit-animation-duration
 - 애니메이션이 실행되는 총 시간을 지정한다. 이 시간 내에 키프레임이 연결되어 애니메이션이 실행된다. (기본 값: 0)
- webkit-animation-iteration-count
 - 애니메이션의 반복횟수를 지정한다. 기본적으로 키프레임은 시작~종료까지 한번만 실행된다. 두 번 이상 실행하고 싶으면 이 속성에 반복하고자 하는 횟수를 지

정하면 된다. 만일 계속 실행(무한 실행)하고 싶을 경우 infinite를 지정하면 된다. (기본 값: 1)

- webkit-animation-direction
 - 키프레임의 연결 방향을 지정한다. 기본적으로 키프레임은 from(혹은 0%) ~ to(혹은 100%) 방향으로 연결된다. 만일 역방향으로의 연결을 원한다면 alternate를 지정하면 된다. (기본 값: normal)
- webkit-animation-timing-function
 - 키프레임간 변화의 정도를 지정한다. (기본 값: ease)
- webkit-animation-play-state
 - 애니메이션의 실행 상태를 지정한다. running 와 paused 값이 있다. (기본값: running) 애니메이션 실행 도중 일시정지, 재시작 시킬 경우 이용한다.
- webkit-animation-delay
 - 애니메이션이 시작되기 전 대기시간을 지정한다. (기본 값: 0, 즉시 시작)

1.16 CSS3 2D Transform

2D 트랜스폼 효과를 줄 수 있는 CSS 2D 트랜스폼에 대한 설명은 표 1과 같다.

표 1. CSS 2D Transform

transform	요소를 2D나 3D로 변형하는 속성
transform-origin	transform 속성을 사용할 때 기준축을 지정
transform-style	하위요소에 3D공간의 속성을 물려줄지의 여부를 지정
perspective	요소에 원근법을 적용하여 3D형태로 보여지게 하는 속성
perspective-origin	perspective 속성을 사용할 때 x축과 y축의 기준점을 지정하는 속성
backface-visibility	입체적 요소의 뒷면을 보여지게 할지 안보이게 할지 지정하는 속성

1.17 HTML5 WebGL

브라우저가 플러그인의 도움을 받지 않고 3D 웹 그래픽을 표현하기 위한 OepnGL ES 2.0의 자바스크립트 바인딩이다. 이것을 사용해 하드웨어 가속이 되는 실시간 3D 그래픽을 표현할 수 있다.

```
[Callback] interface WebGLContextAttributes {
    attribute boolean alpha;
    attribute boolean depth;
    attribute boolean stencil;
    attribute boolean antialias;
    attribute boolean premultipliedAlpha;
    attribute boolean preserveDrawingBuffer;
};
interface WebGLObject {
};
... (지면상 생략)

interface WebGLContextEvent : Event {
    readonly attribute DOMString statusMessage;

    void initWebGLContextEvent(DOMString typeArg,
        boolean canBubbleArg,
        boolean cancelableArg,
        DOMString statusMessageArg);
};
```

그림 18. HTML5 WebGL

1.18 WebP

손실 압축 이미지 파일을 위한 이미지 포맷으로 구글이 개발하여, 2010년 9월 30일 공개하였다. 비디오 포맷인 VP8의 파생으로, 멀티미디어를 담는 포맷인 WebM의 자매 프로젝트이다.

- 확장자 : .webp
- MIME 종류 : image/webp
- 매직 : WEBP
- 개발 : 구글
- 파일 포맷 종류 : 손실 압축 이미지 포맷

1.19 HTML5 Media Capture

미디어 캡처는 User Agent에서 device의 microphone과 camera에 접근하도록 한다.

```
[Supplemental]
interface HTMLInputElement {
    attribute DOMString capture;
};
[NoInterfaceObject]
interface MediaFileData {
    attribute DOMString codecs;
    attribute unsigned long bitrate;
    attribute unsigned long height;
    attribute unsigned long width;
    attribute float duration;
};
```

```
[NoInterfaceObject]
interface MediaFileDataError {
    const unsigned short UNKNOWN_ERROR = 0;
    const unsigned short TIMEOUT_ERROR = 1;
    readonly attribute unsigned short code;
};
```

그림 19. HTML5 Media Capture

1.20 Microdata

마이크로데이터는 웹 페이지 콘텐츠에 시멘틱 요소를 삽입하기 위한 규약이다. 검색엔진, 웹 데이터 수집기, 브라우저는 웹 페이지에서 마이크로데이터를 추출하고 처리한 후, 보다 향상된 사용자 경험을 제공하는데 사용할 수 있다. 마이크로데이터는 아이템과 프로퍼티에 값을 할당하기 위한 네임벨류 쌍을 지원하는 규약 단어를 사용한다. 마이크로데이터에는 아래와 같이 5개의 attribute가 있다.

- **itemscope** - 아이템을 생성하고 이 요소에 관한 정보를 담고 있는 자손요소를 가리킨다.
- **itemtype** - 아이템과 프로퍼티의 문맥을 설명하는 유효한 규약 단어의 URL이다.
- **itemid** - 아이템만의 식별자를 가리킨다.
- **itemprop** - 특정 아이템의 프로퍼티 값을 지닌 태그를 가리킨다. 프로퍼티의 이름과 값은 아이템 규약 단어에 의해 설명하게 된다. 일반적으로 프로퍼티 값을 문자열인 값으로 구성되지만 요소를 사용하는 URL과 href 속성, 이미지 요소, src 속성 또는 외부 리소스를 연결하거나 삽입하는 요소도 사용할 수 있다.
- **itemref** - itemscope와 관련된 아이템의 자손이 아닌 프로퍼티를 가리킨다. 문서에서 추가적인 프로퍼티를 지닌 요소의 id 리스트로 제공된다.

2. 브라우저 호환성 유지 기술 분석

브라우저 호환성 유지 필수 항목들을 기초로 호환성을 유지하기 위한 기술이 어떤 것이 있는지 확인한다. 각 항목 별로 대응 가능한 호환성 라이브러리 조사 결과이다.

2.1 HTML5 Audio

- audio.js[5]
 - audio.js는 HTML5 <audio> tag를 쉽게 이용 가능하게 하는 자바스크립트 라이브러리 이다.
- jPlayer[6]
 - jPlayer는 jQuery 플러그인이며 그 기능은 다음과 같다.
 - ① 웹페이지에서의 미디어 파일 제어
 - ② HTML과 CSS를 이용한 미디어 플레이어 생성 및 스타일 입히기
 - ③ jQuery 프로젝트를 통한 오디오, 비디오 추가
 - ④ HTML5를 이용한 지원 가능한 디바이스 추가
 - ⑤ 플래쉬 fallback을 이용한 구형 브라우저 지원
 - ⑥ 자바스크립트 API를 통해 서비스 중인 웹사이트의 미디어 파일 제어
- File-Formats[7]
 - 다른 형식의 파일 형식을 지원하는 자바스크립트 라이브러리
- XAudioJS[8]
 - XAudioJS는 PCM 형식의 오디오 샘플을 작성하는데 필요한 최소한의 크로스브라우저 API이다. 또한 native 오디오 샘플을 자바스크립트를 이용하여 오디오 시스템에 출력 가능하게 한다.

2.2 HTML5 Button

- html5support[9]
 - html5support는 html5을 지원하지 않는 브라우저를 대상으로 몇몇 html5 요소를 지원한다. 또한 placeholder 속성과 autofocus 속성을 지원하며 type="password" 필드를 지원한다.

2.3 HTML5 datalist

- jquery.datalist.js[10]
 - jquery.datalist.js는 html5 datalist 지원을 위한 크로스브라우저 플러그인이다.
- Relevant-Dropdowns[11]
 - Relevant-Dropdowns는 jQuery와 Modernizr에 의존한 HTML5 datalist 폴리필이다.

- webshim[12]

- Webshims 라이브러리는 개발자가 모든 브라우저에서 현대적으로 크로스 브라우저에 초점을 맞추어 강력한 코드를 작성할 수 있도록 하며 안정적인 HTML5 기능의 정확한 구현에 초점을 맞추며 모듈러 기능에 기반한 polyfill 로딩 라이브러리이다. Webshims 라이브러리는 jQuery와 Modernizr를 기반으로 만들어졌다.

2.4 HTML5 form

- jquery.html5form[13]
 - jquery.html5form은 HTML5을 추가적인 자바스크립트 없이 검증한다. jquery.html5form는 단지 HTML5 문법에 맞추어 form을 생성하면 자동으로 알아서 동작한다. 또한 별도의 클래스가 필요하지 않는다.

2.5 HTML5 menu

- contextMenu[14]
 - contextMenu는 context menu에 대한 관리 도구이다. context menu는 몇백가지의 html5 context menu 요소들을 가지고 있는 응용프로그램을 위하여 만들어졌기 때문에 빠르고 사용메모리 또한 작다. contextMenu는 실제 markup을 사용하지 않고도 context menu 등록을 지원한다. (예 : \$.contextMenu generates DOMElements)

2.6 HTML5 ol

reverse ordered list에 대하여 지원하는 폴리필이다.[15][16]

2.7 HTML5 progress

HTML5 progress 요소를 지원하는 폴리필이다.[17]

2.8 HTML5 rt

CSS를 이용한 HTML5 Ruby annotation이다.[18]

2.9 HTML5 video

- SublimeVideo[19]
 - SublimeVideo는 어떠한 자바스크립트 라이브러리를

이용하지 않은 순수한 자바스크립트 비디오 재생기이다. SublimeVideo는 다른 자바스크립트를 사용하는 웹서비스에서 상당히 안정스럽게 서비스를 할 수 있다.

- Video.js[20]
 - Video.js는 HTML5 video를 지원하기 위해 만든 자바스크립트와 CSS의 모음 라이브러리이다. Video.js는 “HTML5 Video Player”로 알려져있다. Video.js는 또한 크로스 브라우저를 위한 HTML5/CSS 공통 컨트롤 스킨을 제공하며, 전체화면과 서브타이틀 같은 기능을 지원하고, HTML5가 지원하지 않는 플래시 fallback같은 기술요소를 지원한다. 또한 video를 지원하는데 필요한 자바스크립트 API를 제공한다.

2.10 HTML5 history

History.js는 모든 브라우저에서 History/State API(pushState, replaceState, onPopState등)을 훌륭하게 지원한다.[21]

2.11 HTML5 file

- File Reader[22]
 - jDataView는 모든 브라우저에서 바이너리 파일을 읽을 수 있는 방법을 제공한다. DataView 표준을 따르며 표준이상의 실험적인 기능도 포함되어 있다.
- File Saver[23]
 - FileSaver.js는 원래 지원하지 않는 W3C의 saveAs() 인터페이스 구현체이며 클라이언트측의 파일 저장 솔루션이며, 파일을 생성하거나 저장하는 기능을 가진 웹앱에 대한 최적의 솔루션이다.

2.12 HTML5 WebSocket

- SockJS[24]
 - SockJS는 웹소켓 같은 객체를 지원하는 자바스크립트 라이브러리이다. SockJS는 크로스 브라우저를 지원하며 자바스크립트 API를 제공하며 전이중방식을 지원하고 웹서버-클라이언트 브라우저간 크로스 도메인 통신을 지원한다.

- Socket.IO[25]
 - Socket.IO는 모든 브라우저나 모바일 디바이스에 리얼타임 응용프로그램을 만들 수 있도록 기능을 제공한다. Socket.IO는 이기종간 통신을 지원하는 자바스크립트 라이브러리이다.

2.13 CSS3 2D Transform

인터넷 익스플로러를 포함한 크로스브라우저 CSS Transform이다.[26]

3. 브라우저 호환성 유지 기술 분석 사례

브라우저 호환성 라이브러리 항목들 중에서, 대표적인 분석 내용은 다음과 같다. 우선 먼저 해당 feature (예. HTML5 Form)를 지원하는 지 확인하는 단계가 있다. 즉, element를 실제로 create해보고 특정 값을 set한 다음, set한 것이 제대로 되었는지 바로 확인한다. 이때 브라우저가 잘 동작하면 지원, 그렇지 않으면 미지원으로 판단한다.

아래는 실제로 구현한 자바스크립트 소스으로써, canvas를 지원하는지 테스트하는 함수 구현체이다.

```
tests['canvas'] = function() {
  var elem = document.createElement( 'canvas' );
  return !(elem.getContext && elem.getContext('2d'));
};
```

그림 20. Form을 지원하는지에 대한 판단 함수

아래와 같이 확장도 가능하다.

```
Modernizr.addTest('mediaqueries',
  Modernizr.mq('only all'));
```

그림 21. Form을 지원하는지에 대한 확장 함수

각 모듈(예. canvas 등)의 지원뿐만 아니라 내장 method까지도 확인해야 하므로, 추가 지원 체크가 필요한데, 기 구현된 부분을 확장하여 사용한다. 즉, 아래와 같이 fillStyle, fillRect를 확인할 수 있는데, 수동으로 눈으로 확인하는 방법도 있지만, 각 method 호출시 if(ctx.fillStyle === undefined) 과 같이 확인하거나, try ~ catch 구문을 활용하여 체크할 수 있다.

```

var canvas = document.getElementById("canvas");
if (canvas.getContext) {
    var ctx = canvas.getContext("2d");

    ctx.fillStyle = "rgb(200,0,0)";
    ctx.fillRect (10, 10, 55, 50);

    ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
    ctx.fillRect (30, 30, 55, 50);
}
    
```

그림 22. 각 모듈 지원 판단 함수

지원하지 않는 feature에 대해서 어떻게 처리하는가에 대한 것은 다음과 같다.

첫 번째로는 default로 탑재된 것으로 html요소에 global object로 Modernizr을 추가해서 실제로 property가 있는지 확인하는 방법이다. 두 번째는 선택적인 방법으로 기본적으로는 탑재되지 않고, <http://www.modernizr.com/download/>에서 사용자가 원하는 부분을 선별하면 그것에 대한 폴리필 코드를 추가하여 지원하지 않는 브라우저에서도 지원이 가능하도록 해주는 부분이다.

III. SVG를 Canvas로 변환 기술 분석

SVG(Scalable Vector Graphics)는 2차원 벡터 그래픽을 표현하기 위한 XML기반의 파일 형식으로, 1999년 W3C(World Wide Web Consortium)의 주도하에 개발된 오픈 표준의 벡터 그래픽 파일 형식이다. SVG 형식의 이미지와 그 작동은 XML 텍스트 파일들로 정의 되어 검색화, 목록화, 스크립트화가 가능하며 필요하다면 압축도 가능하기 때문에 모바일 장치와 잘 어울린다. SVG 형식의 파일은 SVG기반의 전문 그래픽 편집 프로그램을 사용하여 편집이 가능하며 XML 파일로 되어 있으므로 문서 편집기로도 편집이 가능하다.

현재 마이크로소프트의 인터넷 익스플로러를 제외한 대부분의 주요 웹 브라우저들은 SVG를 지원한다. 인터넷 익스플로러에서는 SVG 파일을 보기 위해 별도의 플러그인을 수동으로 설치하여야 하며, 그렇지 않은 경우에는 웹 페이지 제작자가 구글 코드에서 개발 중인 자바스크립트 라이브러리, SVG Web을 웹 페이지 코드에 포함시켜야 한다.

모바일 단말에서 SVG 를 지원하기 위해서는 약 1MB 정도의 코드가 추가로 필요하여 기본 기능으로 탑재하기에는 부담이 크다. 예를 들면, 안드로이드의 경우 SVG를 지원하

는 코드가 포함되어 있지만 상용 버전에서는 이 기능을 포함하지 않고 있다. 따라서 SVG 미지원 단말에서도 이를 포함하는 콘텐츠를 출력하기 위해서는 SVG를 파싱하여 렌더링할 수 있는 기능이 필요하다.

브라우저가 Canvas를 지원하는 경우, 자바스크립트를 사용하여 SVG 파일을 파싱 후 렌더링 하는 기술이 개발되고 있다. SVG2Canvas 솔루션인 canvg(<http://code.google.com/p/canvg/>)는 Google code에서 오픈 소스로 진행되는 프로젝트로, SVG 파일에 대한 URL이나 SVG 파일 자체를 텍스트 형식으로 가져와서 자바스크립트를 사용하여 파싱한 후에 Canvas 에 결과를 렌더링 한다. 렌더링 속도는 내장 SVG 엔진과 비슷하다.

IV. 그래픽 요소 벤치마크

그래픽 요소 벤치마크 목적은 N-스크린 각각의 단말에서 그래픽 요소의 기본 성능을 파악하고, 그래픽 요소의 각 항목이 성능에 어떤 영향을 미치는지 파악하는 것이다. 그래픽요소 벤치마크 항목은 표2와 같다.

그래픽요소 벤치마크 측정 방법은 크게 루프문을 활용한 측정하는 방법과 타이머를 활용한 측정 방법으로 나누어진다. Loop 문을 활용한 측정 방법은 그래픽이 끊어지면서 출력되는 특징이 있다. 동기적으로 Update할 경우, Browser Engine에서 direct로 화면 업데이트를 하지 않는 문제로 인하여 정확한 성능을 측정할 수 없는 단점이 있다. Timer를 활용한 측정 방법은 그래픽이 부드럽게 출력되는 특징이 있다. 비 동기적으로 갱신할 경우, Browser Engine이 화면 업데이트 할 시간을 가질 수 있으므로 측정의 기준으로 삼을 수 있다.

표 2. 그래픽요소 벤치마크 항목

Test	Objective
CSS	div, image size가 속도에 영향을 주는지 검증
	div, image 개수가 속도에 영향을 주는지 검증
	div transform-style이 속도에 영향을 주는지 검증
	image 종류가 속도에 영향을 주는지 검증
	2D transforms와 3D transform의 속도 비교 CSS Transition과 CSS Animation의 성능 비교
SVG & Canvas	SVG 또는 Canvas size, shape size가 속도에 영향을 주는지 검증
	Shape의 개수가 속도에 영향을 주는지 검증
	SVG filter 또는 Canvas style이 속도에 영향을 주는지 검증
	SVG와 Canvas의 성능 비교

V. 결론

N-스크린 콘텐츠를 제작할 때 호환성 라이브러리, SVG 변환 기술, 그래픽요소 성능 가이드를 통하여 다양한 N-스크린 콘텐츠의 호환성을 검증할 수 있는 기술을 살펴보았다. 브라우저의 호환성 유지 방법, SVG를 Canvas로 변환하는 방법 및 그래픽 요소 벤치마크 자료를 활용한다면 실제 N-스크린 콘텐츠 개발에 많은 도움을 줄 것으로 기대한다. 또한, 최적의 성능을 가진 N-스크린 콘텐츠 개발의 핵심 자료로 사용할 수 있을 것으로 기대한다.

참고문헌

- [1] <http://addyosmani.com/blog/writing-polyfills/>
- [2] <http://www.w3.org/TR/SVG/>
- [3] <http://www.adobe.com/kr/enterprise/svg.html>
- [4] 김자용, 이승희, 최진영, “웹 표준 및 접근성을 강화한 시스템 운영방안(A System management plan for enhancing web standard and accessibility),” 한국정보과학회 2010 한국컴퓨터 학술발표논문집 제37권 제2호(C), pp. 160-163, Nov. 2010.
- [5] <http://kolber.github.com/audiojs>
- [6] <https://github.com/happyworm/jPlayer>
- [7] <https://github.com/bebraw/jswiki/wiki/FileFormats>
- [8] <https://github.com/grantgalitz/XAudioJS>
- [9] <https://github.com/amiel/html5support>
- [10] <https://github.com/miketaylr/jquery.datalist.js>
- [11] <https://github.com/chriscoyier/RelevantDropdowns>
- [12] <http://afarkas.github.com/webshim/demos/>
- [13] <http://www.matiasmancini.com.ar/jquery-plugin-ajax-form-validation-html5.html>
- [14] <https://github.com/medialize/jQuery-contextMenu>
- [15] <https://github.com/impressivewebs/HTML5-Reverse-Ordered-Lists>
- [16] <https://gist.github.com/1671548>
- [17] <https://github.com/LeaVerou/HTML5-Progress-polyfill>
- [18] <http://www.useragentman.com/blog/2010/10/29/cross-browser-html5-ruby-annotations-using-css/>
- [19] <http://sublimevideo.net/>
- [20] <http://videojs.com/>
- [21] <https://github.com/balupton/history.js>
- [22] <https://github.com/vjeux/jDataView>
- [23] <https://github.com/eligrey/FileSaver.js>
- [24] <https://github.com/sockjs/sockjs-client>
- [25] <http://socket.io/>
- [26] <http://www.useragentman.com/blog/2010/03/09/cross-browser-css-transforms-even-in-ie/>

※ 이 논문은 한국콘텐츠진흥원의 “2011년 콘텐츠산업기술지원사업”의 지원으로 연구된 결과입니다.

저 자 소 개



이 정 표

1998: 인하대학교
물리학과 이학사.
2002-2008: Teleca Korea
수석연구원.
2010: Teleca Korea 기술이사
현 재: 케이티하이텔(주)
플랫폼사업부 부장
관심분야: Web & 모바일 플랫폼,
모바일컴퓨팅,
클라우드컴퓨팅,
N-Screen 서비스



이 원 주

1989: 한양대학교
전자계산학과 공학사.
1991: 한양대학교
컴퓨터공학과 공학석사.
2004: 한양대학교
컴퓨터공학과 공학박사.
현 재: 인하공업전문대학
컴퓨터정보과 부교수.
관심분야: 병렬처리시스템, 성능분석,
모바일컴퓨팅,
클라우드컴퓨팅,
N-Screen 서비스



김 민 태

1991: 정석항공공업고등학교
항공 전자과.
1995~1997: 한국신용평가
1997~1999: 보앤엘소프트
개발 팀장
1999~2000: 한국기술표준원
연구원
2000~2004: (주)투비웨이
개발 선임
2004~2011: (주)이미지클릭
개발 팀장
현 재: 케이티하이텔 개발실
iOS팀 및 Web플랫폼
Lab 팀장
W3C HTML5 KIG UI
그룹장
관심분야: Web & 모바일 플랫폼,
모바일컴퓨팅,
클라우드컴퓨팅,
N-Screen 서비스