Icon Standardization For Web Applications

Abstract

This research is derived from my talk at ArabNet conference held in Beirut on 2013-03-20.

The aim of this research is to develop standards for representing icons in web applications. This means giving proposed icons in this research addresses in Miscellaneous Symbols and Arrows and Miscellaneous Technical unicode blocks. e.g.:

- 2BEA = RIGHT-POINTING MAGNIFYING GLASS WITH PLUS SIGN = Zoom in
- 2BEB = RIGHT-POINTING MAGNIFYING GLASS WITH MINUS SIGN = Zoom out
- 23FB = FOUR ANGULAR ARROWS FACING OUT = Full-screen
- 23FC = FOUR ANGULAR ARROWS FACING IN = Exit full-screen
- ...

Methedology

- First of all, I will show the solutions currently being used to display icons in web applications:
  - Whether using the usual image files,
  - Or using the embedded-fonts technology.
- Then I will form a model for a proposed website, in order to summarize the icons needed for most web applications.
  - In light of this model, I will go to propose a set of recommendations to:
    - Unicode organization to determine Unicode icons missing in the current version of Unicode,
    - and to W3C - The World Wide Web Consortium, to visualize a view for representing icons using CSS.

Using Regular Image Files

The most popular solution used to display icons in web applications is called "CSS Sprites", which is to collect the various needed icons -of all sizes and colors- in a single regular image file. Here is a sample of Google icons stored in a single regular image file. Notice that icons are spreaded over the image canvas randomly.
Using The Embed Fonts Technology

The embed fonts technology has paved a new way for web developers to represent icons using characters in the embedded fonts instead of the regular image formats used usually in the web, such as PNG, GIF... etc, which represents an important step for icon addressing in software industry generally, and in web development world particularly.

This step coincided with another important step as important as its previous one, which is the adoption of Unicode 6.0 for a broad spectrum of visual symbols (Emoji) and embedding it in its tables.

> Emoji are the ideograms or smileys used in Japanese electronic messages and webpages, whose use is spreading outside Japan. Originally meaning pictograph, the word emoji literally means “picture” + “character”. [Wikipedia, 2014](https://en.wikipedia.org/wiki/Emoji)

This step might lead us to a completely different usage of Emoji; providing a stable standard for addressing UI icons in software engineering. According to this version of Unicode, each popular UI icon had a stable address in Unicode blocks. For example; the muted speaker has the address 1F507, and the magnifying glass has the address 1F50D... and so on.
RichStyle font is an attempt to implement the idea of icons-addressing in the web pages through the Unicode 6.0 and above, but this project revealed a lack of some icons and visual symbols that are not addressed yet by Unicode. These icons are the ones that have a label in the preview page of RichStyle font.

Recomendations for Unicode: Missing Icons

In order to clarify this limitation I will try to develop a model for a generic web site covers the most popular requirements for a modern web site.
Notice that:

- Every single class in this model needs an icon.
- Every single operation in each class needs an icon too.
- Attributes do not need icons.

In the light of this model, and comparing to the icons supported in Unicode 7.0 beta, web developer
can use the following Unicode icons:

### Table 1 - Available web Emoji icons for Unicode 7.0

<table>
<thead>
<tr>
<th>Item</th>
<th>Code</th>
<th>Character</th>
<th>Names</th>
<th>Alias Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>about class</td>
<td>1F6C8</td>
<td><img src="image" alt="i" /></td>
<td>CIRCLED INFORMATION SOURCE</td>
<td></td>
</tr>
<tr>
<td>audio class</td>
<td>1F3B5</td>
<td><img src="image" alt="♫" /></td>
<td>MUSICAL NOTE</td>
<td>music, being in good mood, audio clip</td>
</tr>
<tr>
<td>bookmark operation</td>
<td>1F516</td>
<td><img src="image" alt="buat" /></td>
<td>BOOKMARK</td>
<td></td>
</tr>
<tr>
<td>close operation</td>
<td>1F5D9</td>
<td><img src="image" alt="close" /></td>
<td>CANCELLATION X</td>
<td>close</td>
</tr>
<tr>
<td>contact-us class</td>
<td>2709</td>
<td><img src="image" alt="✉️" /></td>
<td>ENVELOPE</td>
<td></td>
</tr>
<tr>
<td>documentation class</td>
<td>1F4D6</td>
<td><img src="image" alt="📖" /></td>
<td>OPEN BOOK</td>
<td>read operator's manual</td>
</tr>
<tr>
<td>document class</td>
<td>1F5CE</td>
<td><img src="image" alt="document" /></td>
<td>DOCUMENT</td>
<td></td>
</tr>
<tr>
<td>downloads class</td>
<td>2B8B</td>
<td><img src="image" alt="⬇️" /></td>
<td>DOWNWARDS BLACK CIRCLED WHITE ARROW</td>
<td></td>
</tr>
<tr>
<td>e-book class</td>
<td>1F56E</td>
<td><img src="image" alt="📖" /></td>
<td>BOOK</td>
<td></td>
</tr>
<tr>
<td>edit operation</td>
<td>270E</td>
<td><img src="image" alt="_pen" /></td>
<td>LOWER RIGHT PENCIL</td>
<td></td>
</tr>
<tr>
<td>flip-horizontal operation</td>
<td>2B0C</td>
<td><img src="image" alt="↔️" /></td>
<td>LEFT RIGHT BLACK ARROW</td>
<td>flip-horizontal</td>
</tr>
<tr>
<td>flip-vertical operation</td>
<td>2B0D</td>
<td><img src="image" alt="↕️" /></td>
<td>UP DOWN BLACK ARROW</td>
<td>flip-vertical</td>
</tr>
<tr>
<td>go-top operation</td>
<td>2B89</td>
<td><img src="image" alt="⬆️" /></td>
<td>UPWARDS BLACK CIRCLED WHITE ARROW</td>
<td></td>
</tr>
<tr>
<td>home class</td>
<td>1F3E0</td>
<td><img src="image" alt="🏠" /></td>
<td>HOUSE BUILDING</td>
<td></td>
</tr>
<tr>
<td>image class</td>
<td>1F5BC</td>
<td><img src="image" alt="🖼️" /></td>
<td>FRAME WITH PICTURE</td>
<td>art, graphic document</td>
</tr>
<tr>
<td>line-wrap operation</td>
<td>2B92</td>
<td><img src="image" alt="⏎" /></td>
<td>NEWLINE LEFT</td>
<td></td>
</tr>
<tr>
<td>log-in operation</td>
<td>1F512</td>
<td><img src="image" alt="🔒" /></td>
<td>LOCK</td>
<td></td>
</tr>
<tr>
<td>log-out operation</td>
<td>1F513</td>
<td><img src="image" alt="🔓" /></td>
<td>OPEN LOCK</td>
<td></td>
</tr>
<tr>
<td>plain-text class</td>
<td>1F5B9</td>
<td><img src="image" alt="📝" /></td>
<td>DOCUMENT WITH TEXT</td>
<td></td>
</tr>
</tbody>
</table>
Underlined term means proposal. Notice also that I redefined \( 1 \text{F5BB} \) as an icon for presentation document.

Whilst, web developer still not find matching unicode addresses for the following classes and operations, which means it need to be added in the official/final Unicode 7.0 release:

| Table 2 - Proposed web Emoji icons for Unicode 7.0 |
|---------|-------------|-------------------------------|---------------------|
| Item    | Proposed Code | Character                     | Names                            | Alias Names                  |
| full-screen operation | 23FB | 🔥🔥🔥🔥 | FOUR ANGULAR ARROWS FACING OUT | full-screen                  |
| exit-full-screen operation | 23FC | 🔥🔥🔥🔥 | FOUR ANGULAR ARROWS FACING IN  | exit full-screen              |
| package class | 2BE0 | 🔒 | TIED FOLDER | archive, |
Recomendations for W3C: Representing Icons Using CSS

W3C standards stated that icons should be represented using the following syntax:

```html
<menu>
  <menuitem icon='page-about.png'/>
  <menuitem icon='domain-settings.png'/>
</menu>
```

Unfortunately, this standard will not allow you to implement the popular ways used nowadays to represent icons, whether through CSS Sprites or embedded fonts technology, it’s a backward movement, and it means representing each icon using an independent image file, and therefore fatiguing the network with too many request for downloading a large amount of images.

As an alternative, I suggest representing icons using CSS pseudo element, not as an HTML element’s
attribute. The reason for this is that icon—even though it represents an attribute— but:

- First of all; it’s a presentation attribute, and therefore must be represented using CSS rather than HTML,
- Secondly, it’s a rich attribute; an attribute needs attributes. These kind of attributes are usually represented using something called Pseudo Element.

practically; in the HTML page, we’ll use the element MenuItem with a class name, and in CSS file we’ll use a pseudo element called :icon, to be able to adjust all the presentation attributes of this icon, just like content, font name, font size, and position.

```html
<menu>
  <menuitem class='page-about'/>
  <menuitem class='domain-settings'/>
</menu>
```

```
.page-about:icon {
  content: char(ℹ);
  font: 10pt RichStyle;
  icon-position: top;
}
.domain-settings:icon {
  content: char(⚙);
  font: 10pt RichStyle;
  icon-position: top;
}
```

**Icon Standardization for Desktop Applications**

A few years ago, Tango Icons project have tried to set a standard for icons naming for Linux, in order to be addressed in a standard way. However, I think that Unicode 6.0+ now represents a reliable alternative to be adopted as a standard for addressing icons within the software process.

![Figure 4 - Unicode 6.0 icons naming vs. Tango icons naming](image)

Here is an express view of proposed icons for desktop and mobile icons, inspired by Unicode 7.0 Beta itself.
**Conclusion**

- Some Emoji icons for web environment need redefining, as shown in Table 1.
- Some Emoji icons for web environment are missed, and need to be added to Unicode as shown in Table 2.
• Some Emoji icons for desktop and mobile environments are missed, and need to be added to Unicode as shown in Figure 5.
• For W3C, icons should be defined using CSS; not HTML, as Pseudo Elements, i.e. :icon{...}

References

• Unicode 7.0 Beta Symbols:
  - Supplemental Arrows-C: http://www.unicode.org/charts/PDF/Unicode-7.0/U70-1F800.pdf

• RichStyle font (list of the icon designers including The Noun Project designers): http://richstyle.org/font.php

© 2006-2014 Mohammad Anas Ramadan