Reacting to the changes in screen size in Symbian Web Runtime

Overview

Because of the vast amount of different screen sizes on Nokia devices, it is important that the developer pays attention to the scalability of his or her widget. There are a couple of things which need to be taken into account in this respect. The first is detecting when the screen size changes (for more information about this, see Detecting orientation changes in Symbian Web Runtime). The second is what to do when the screen size changes. This snippet covers the latter aspect.

The technique presented in this snippet is enabling and disabling style sheets dynamically depending on the resolution in question. For example, if the resolution is detected to be nHD portrait (360x640), then the CSS file for this resolution is enabled and all others are disabled.

CSS files

First, let's define a style sheet that is independent of the resolution:

```
body {
    background-repeat: no-repeat;
}

h1 {
    text-align: center;
}
```

Then, let's create a style sheet that contains styles for nHD resolution in portrait orientation (360x640 pixels). As an example, this CSS file sets the background image of the body element to a picture corresponding to the resolution in question:

```
body {
    background-image: url("../gfx/nhd_portrait.png");
}
```

Similarly, let's create style sheets for nHD landscape (640x360 pixels), QVGA portrait (240x320 pixels), and QVGA landscape (320x240 pixels) resolutions, as well as for the mini view (the home screen, 91x315 pixels):

```
body {
    background-image: url("../gfx/nhd_landscape.png");
}
```

```
body {
    background-image: url("../gfx/qvga_portrait.png");
}
```

```
body {
    background-image: url("../gfx/qvga_landscape.png");
}
```

```
body {
    background-image: url("../gfx/home_screen.png");
}
```


(C) Copyright Nokia 2014. All rights reserved.

Source

The CSS files can be added to the HTML page as external style sheets by using the `link` element. If they are given a title with the `title` attribute, they become preferred style sheets (yes, there can be more than one of these) and it becomes quite easy to refer to them through JavaScript. You'll see an example of this later on. In addition, if you define the `disabled` attribute for them, they can be disabled and enabled programmatically through JavaScript. This is also shown in practice later on in this snippet. Here are all the style sheets created in the previous chapter:

```
<head>
    <link rel="stylesheet" href="style/persistent.css" type="text/css" />
    <link rel="stylesheet" href="style/nhd_portrait.css" type="text/css"
        disabled="true" title="NHD Portrait" />
    <link rel="stylesheet" href="style/nhd_landscape.css" type="text/css"
        disabled="true" title="NHD Landscape" />
    <link rel="stylesheet" href="style/qvga_portrait.css" type="text/css"
        disabled="true" title="QVGA Portrait" />
    <link rel="stylesheet" href="style/qvga_landscape.css" type="text/css"
        disabled="true" title="QVGA Landscape" />
    <link rel="stylesheet" href="style/home_screen.css" type="text/css"
        disabled="true" title="Home Screen" />
</head>
```

The style sheet without the `title` attribute, `persistent.css`, is a so-called persistent style sheet. It will never be disabled. From the others, the one corresponding to the current screen size will be enabled and all others will be disabled. Here is the code that iterates through the style sheets and checks whether any of them corresponds to the title received as a parameter:

```
// Sets the active stylesheet for the HTML document according to the specified
// title.
function setActiveStyleSheet(title) {
    var stylesheets = document.getElementsByTagName("link");
    for (var i = 0; i < stylesheets.length; i++) {
        var stylesheet = stylesheets[i];
        // If the stylesheet doesn't contain the title attribute, assume it's
        // a persistent stylesheet and should not be disabled
        if (!stylesheet.getAttribute("title")) {
            continue;
        }
        // All other stylesheets than the one specified by "title" should be
        // disabled
        if (stylesheet.getAttribute("title") != title) {
            stylesheet.disabled = true;
        } else {
            stylesheet.disabled = false;
        }
    }
}
```

The above code is called whenever the screen size changes (see also Detecting orientation changes in Symbian Web Runtime):

```
// Called when the window is resized
function windowResized() {
    var prevResolution = resolution;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    // Detect the new resolution
    detectResolution();

    if (resolution == RESOLUTION_NHD_PORTRAIT) {
        setActiveStyleSheet("NHD Portrait");
    } else if (resolution == RESOLUTION_NHD_LANDSCAPE) {
        setActiveStyleSheet("NHD Landscape");
    } else if (resolution == RESOLUTION_QVGA_PORTRAIT) {
        setActiveStyleSheet("QVGA Portrait");
    } else if (resolution == RESOLUTION_QVGA_LANDSCAPE) {
        setActiveStyleSheet("QVGA Landscape");
    } else if (resolution == RESOLUTION_HOME_SCREEN) {
        setActiveStyleSheet("Home Screen");
    } else if (resolution == RESOLUTION_UNDEFINED) {
        // TODO: Error handling
    }
}
```

Postconditions

The CSS file corresponding to the current resolution is enabled and all others are disabled.

See also

- Detecting orientation changes in Symbian Web Runtime

Supplementary material

This code snippet is part of the stub concept, which means that it has been patched on top of a template application in order to be more useful for developers. The version of the WRT stub application used as a template in this snippet is v1.2.

- The patched, executable application that can be used to test the features described in this snippet is available for download at Media:reacting to orientation changes.zip.
- You can view all the changes that are required to implement the above-mentioned features. The changes are provided in unified diff and colour-coded diff (HTML) formats in Media:reacting to orientation changes.diff.zip.
- For general information on applying the patch, see Using Diffs.
- For unpatched stub applications, see Example app stubs with logging framework.

(C) Copyright Nokia 2014. All rights reserved.