Coding Nokia Asha web apps for screen orientation

This article shows how to display your application in landscape on those Nokia Asha devices that offer QVGA landscape screens.

Introduction

Most Nokia Asha devices that run Series 40 web apps offer a portrait screen, in QVGA (240 x 320-pixel) resolution. However, there are a small number of Nokia Asha devices that offer QVGA landscape orientation screens as well. If you do nothing, your web app continues to display in portrait format on the landscape screen, with white space either side of the content and a scroll bar.

Fortunately there is a simple solution, which takes advantage of the fact that Nokia Asha devices don't offer dynamic orientation switching.

Step 1: Use relative formatting

For any content, such as tables, where you have the ability to use relative (percentage) size controls do so. This will mean the content automatically shrinks or stretches to fit the screen width.

Step 2: Create CSS definitions for landscape and portrait formats

Where you don't have the option to set the size of your content relatively, you will need to create CSS definitions to control size. In the code example we are controlling the size of images, so in our CSS for portrait screens

```
.imgSz {
 width:200px;
 height:125px;
}
```

while in the landscape we have

```
.imgSz {
 width:250px;
 height:156px;
}
```

enabling the image to be displayed across more of the screen width, thus making better use of the landscape screen.

Step 3: Create code to query the screen size and set the CSS file to use

Now we need to set the web app to use the CSS for the devices orientation. As Nokia Asha devices don't allow for the dynamic switching of screen orientation, we only need to do this once. The code is simple; it queries the screen object and reads the width property. Next the code tests the screen width (anything greater than 240 indicates landscape orientation). Based on the result a <link> is written to the web app's document for the CSS file defining the layout for the detected screen orientation, as shown in the following code.

```
/**
 * Selecting a CSS file based on screen width
 */
if (screen.width > 240) {
 document.write('<link rel="stylesheet" href="basicLandscape.css" type="text/css" />');
}
else {
 document.write('<link rel="stylesheet" href="basicPortrait.css" type="text/css" />');
}
```


(C) Copyright Nokia 2014. All rights reserved.

Step 4: Run the code

Now it's simply a case of executing the JavaScript code from the <head> of the web app's main HTML, as shown below.

```
<head>
 <title>Screen size example</title>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 <!-- Common css between the two orientations -->
 <link rel="stylesheet" href="common.css" type="text/css" />

 <!-- The following script sets and gets the appropriate CSS file for the screen
 orientation -->
 <script type="text/javascript" src="screensize.js"></script>
</head>
```

With the right CSS in use your web app will now display using the full landscape screen

Now you are able to provide your users with an optimal view of your web app, regardless of whether they view it on a landscape or portrait screen.

Media:ScreenSize.wgt(zip file File:ScreenSize.zip with .wgt extension) provides a small example web app containing this code.

For more information on the best practices in coding your Series 40 web app, see Nokia Asha Web Apps Best Practices Guide.

(C) Copyright Nokia 2014. All rights reserved.