Getting Started with HTML5 using BlackBerry WebWorks

Lab # 2: *Geolocation and New UI Possibilities*

The objective of this lab is to introduce some of the new concepts added in HTML5 by creating a WebWorks application for the BlackBerry Smartphone. In this lab, we focus on adding geolocation to the application and improving the user interface.

It is expected that before attempting this lab you have followed the instructions in Lab 1 in order to install the WebWorks SDK and the Ripple emulator.

Using Geolocation Features

- Open the starup file archive Lab2.zip and extract it to a folder of your choice, or use your code from Lab 1 if you have completed it. Next open index.html in your favorite HTML editor.

- We will first be adding and making use of some of the geolocation features now available by finding the current GPS or IP based location, displaying the coordinates and showing that location on an interactive map.

- Start by adding a script tag to import this URL

  http://maps.google.com/maps/api/js?sensor=false

- Then at the bottom of script.js we will add all of the following code to handle the connection to Google Maps and display it in our app.

```
//More Information at: http://html5demos.com/geo
function success(position) {
  var s = document.querySelector('#status');

  if (s.className == 'success') {

    return;
  }

  s.innerHTML = "Current location " +position.coords.latitude+" , "+
position.coords.longitude;
  s.className = 'success';

  var mapcanvas = document.createElement('div');
  mapcanvas.id = 'mapcanvas';
  mapcanvas.style.height = '400px';
  mapcanvas.style.width = '600px';

  document.querySelector('article').appendChild(mapcanvas);

  var latlng = new google.maps.LatLng(position.coords.latitude,
position.coords.longitude);
```

```
  var myOptions = {
    zoom: 15,
    center: latlng,
    mapTypeControl: false,
       navigationControlOptions: {style:
       google.maps.NavigationControlStyle.SMALL},
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var map = new google.maps.Map(document.getElementById("mapcanvas"),
myOptions);

  var marker = new google.maps.Marker({
      position: latlng,
      map: map,
      title:"You are here! (at least within a
"+position.coords.accuracy+" meter radius)"
  });
}

function error(msg) {
  var s = document.querySelector('#status');
  s.innerHTML = typeof msg == 'string' ? msg : "failed";
  s.className = 'fail';
  }
```

- Next is the important part, in index.html, under netstate line, add

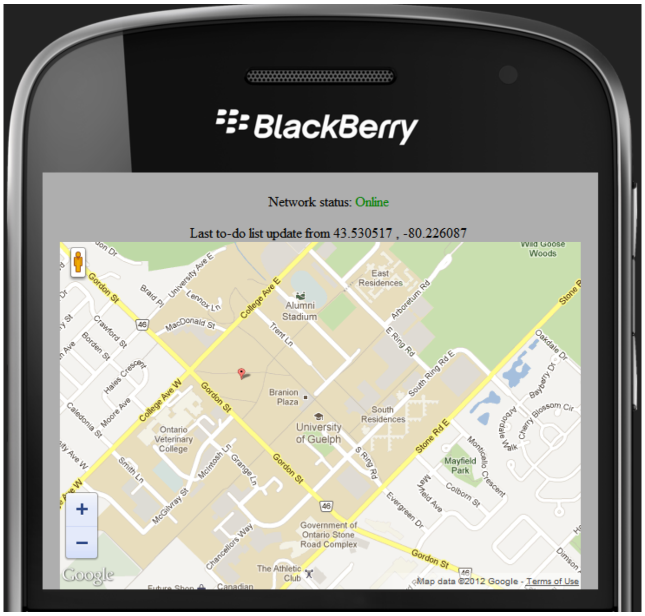
```
<script>

if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(success, error);
} else {
  error('not supported');
}
</script>
```

The `navigator.geolocation.getCurrentPosition(success, error);` line uses the new geolocation functions to obtain the latitude and longitude where the device current is and reports it back to the functions we created earlier to display on the map. When GPS is unavailable, the coordinates may be found based on IP address or other location services. If you go through the JavaScript, you will notice that there is a SPAN that is updated with the status, add this to your index file so that it can correctly identify to the user what the status is and where they last updated their to-do list.

Test this inside Ripple and you should see the map display. If you open the geolocation tab, you can move your location and see it reflected when you reload the application. This is an exciting feature that will let you add location aware services to your web applications. The user is always prompted for permission before this service is allowed.

- Since we are working within WebWorks, in order for this to work on devices the configuration file (config.xml) will need to be modified. Read through https://bdsc.webapps.blackberry.com/html5/documentation/ww_developing/working_with_config_xml_file_1866970_11.html and find the commands to add into the config.xml file to allow reading information from the Google domain and provide access to geolocation functions. Without this, access will be blocked.



Network status: Online

Last to-do list update from 43.530517 , -80.226087

Geolocation Mapping

- Now that you have the coordinates, modify your program so that the current coordinates are saved and displayed as the ‘Last Updated from’ location. This location should be saved between sessions.

Hidden Attributes

- Previously it could be quite difficult to have hidden attributes and often required some JavaScript to hide the values. HTML5 allows a simple addition to hide an element.

- In index.html, after your form closing tag, add a line thanking the user for submitting their information. In the <p> tag, use an ID of ‘thanks’ and set the value ‘hidden = true’. This is all you need now to hide text. If you load your app in ripple you will not see it.

- For your form submit button, set the onclick value to the function formSubmit. Your completed form and thank you text should be similar to as follows

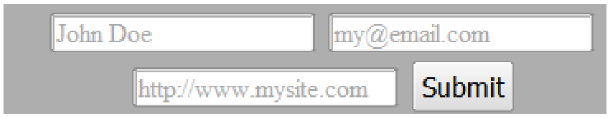
```
<form name="form">
  <input type="text" id="name" name="Name" placeholder= 'John Doe' >
   <input type="email" id="email" name="email" placeholder="my@email.com"><br>
  <input type="URL" id="url" name="website" placeholder=
'http://www.mysite.com' >
   <button type="button" id="submit" onclick = "formSubmit()"> Submit </button>
</form>
<p id="thanks" hidden = true>Thanks for submitting the information.</p>
```

- Now in our script file, we will add a simple function that will clear the form data and unhide our thank you message. In the function, hide the name, email, URL and submit button elements and unhide the thank you message. All of these will be done in a similar manner to this code

  ```
  document.getElementById('thanks').hidden = false;
  ```

  Now when the submit button for the form is clicked, the form disappears and the message is displayed to the user.

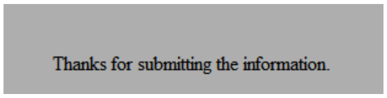
  When testing in Ripple, you will see this feature functioning correctly, however on some BlackBerry device browsers it will not. This is a key note to understanding HTML5, not all features are fully implemented yet on all browsers. Much caution has to be used when deciding to use these features.



John Doe

my@email.com

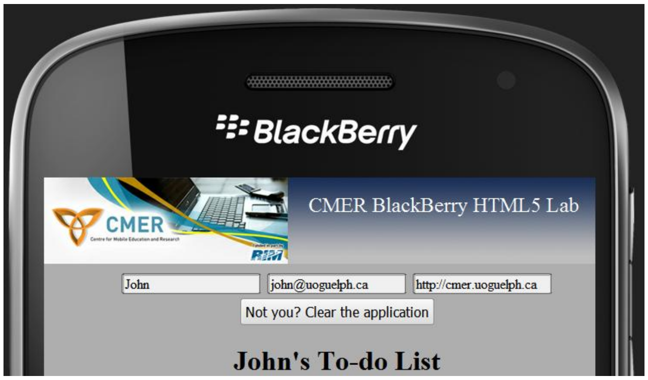
http://www.mysite.com



Thanks for submitting the information.

  When the submit button is clicked, the fields should no longer be visible and the thank you message should appear as seen above.

- Next modify the above code so the name, email and website are stored for the user. When the application is loaded, the correct information should be displayed for the last user including a modified title welcoming them by name. Add a clear button that will allow all information and to-do list items to be cleared from the screen and memory.



CMER BlackBerry HTML5 Lab

John

john@uoguelph.ca

http://cmer.uoguelph.ca

John's To-do List

User Interface Improvements

- We will now modify the to-do list section in order to make it have more of a notepad like feel.

- Wrap your unordered list with 3 successive DIV's. The outer DIV should be called note, then tape followed by text. After closing the text DIV, add a new DIV called line, close it, than do the same called bg. These will provide use some formatting and a notepad like interface for our text.

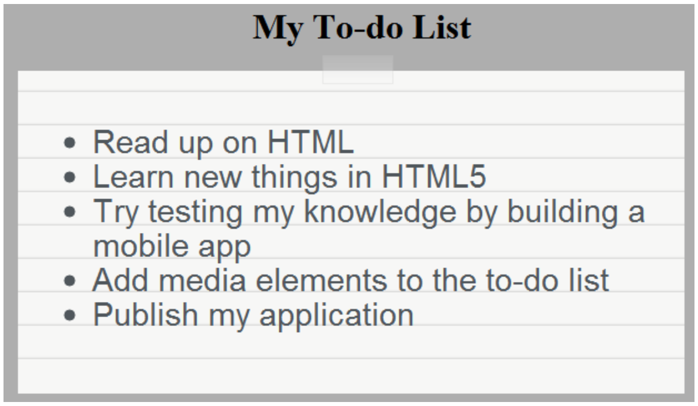
- Add all of the following to the bottom of your style.css file.

```
.note {
	width:600px;
	min-height:120px;
	margin-bottom:30px;
	background: #f7f7f6;
	text-align:left;
	position:relative;
	z-index:999;
}

.tape {
	width:12px;
	height:34px;
	position:absolute;
	background:white;
	border:1px solid #cbcbcb;
	top:-14px;
	left:274px;
	background: -webkit-gradient(linear, 0 bottom, 0 top, from(#fafafa ),
	to(#cfcfcf));
	opacity:.3;
}

.bg {
	width:100%;
	height:100%;
	top:-11px;
	position:absolute;
	z-index:-4;
	background: -webkit-gradient(linear, 0 99%, 0 89%,  from(#d8d8d7) ,
	to(rgba(255,255,255, 0)));
	background-size: 50px 30px;
}
.text {
	width:570px;
	padding:19px 0 27px 27px;
}
.text ul{
	color:#50575c;
	font:29px/31px arial, serif;
}
```

The Note CSS creates the box around the text that we will have with a small tape element created by the tape styling putting a small semi-transparent rectangle as if the note is taped to a wall. You will notice that you will need to adjust the placement of the tape in order for it to to be centered. The WebKit rendering engine inside the BlackBerry browser, introduced in OS 6.0 is largely to do with allowing these intricate interfaces to be created, something previously unavailable to mobile web developers on the platform.



My To-do List

- Read up on HTML
- Learn new things in HTML5
- Try testing my knowledge by building a mobile app
- Add media elements to the to-do list
- Publish my application

The to-do list box should now look similar to the above. The text should remain editable and local storage should also be unaffected.

- Another approach is heavily leveraging JQuery which is the go-to JavaScript Framework for developers that are using these emerging web developing technologies. JQuery uses many HTML5 features in order to provide nicer interfaces. We had already used JQuery in Lab 1 and you are able to use it again here if you so choose. The JQuery website, www.jquery.com provides an API for what it can be used for as well as specific UI and mobile versions.

This is the end of Lab 2, in Lab 3 we will add video and audio support and survey other features.