UsageGuide

How to use the Text 2.0 Framework to create gaze active web apps.

Updated Today (moments ago) by r.biedert

Startup and Running Apps

1. Install the latest Java (v6) and a modern browser (Chrome >9, Firefox 4, Safari 5).
2. Start the **Tracking Server**. This step is optional, but if you skip it, it will take the browser plugin a few seconds search before it gives up and uses the mouse as a default fallback. If you want to use eye tracking, the tracking server has to be running.
3. Start the **Diagnosis**. This step is optional, but recommended to calibrate the eye tracking device and check data arrives. The proper sequence to calibrate is 1) perform *hardware calibration* and 2) perform *local recalibration* with transparency enabled.
4. Open the **Web-App**. After a few moments (100ms up to 5 seconds, depending on whether a tracking server is running) a speech output should be generated saying either "using tracker" or "using mouse". Please note that "using tracker" only indicates that the plugin uses the tracking server and does not state anything about the type of adapter used within the server, i.e., if the adapter is mouse-bound, it will say *tracker* anyway.

Simple Example

Using the framework is straightforward. First, include all required libs (`jquery`, `jquery.timers` and `text20` in our case), then execute the `core.init()` function and start using gaze active handlers like `onFixation`, `onGazeOver` and so forth:

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.timers.js"></script>
<script type="text/javascript" src="text20.js"></script>

<script type="text/javascript">
    text20.core.init()
</script>

<body><div onFixation="alert('Hello again')">Hello World</div></body>
```

Supported Handlers

Here is a brief list what handlers we support and how they work. Inside the attribute you may use `this` to access the annotated element.

```
<!-- This handler is called every time ... a fixation occurs on the annotated element -->
<div onFixation="alert('Fixation on ' + this)" ></div>

<!-- ... the user looks on the element  -->
<div onGazeOver="..." ></div>

<!-- ... the user away from the element  -->
<div onGazeOut="..." ></div>

<!-- ... continuous reading behavior is detected  -->
<div onPerusal="..." ></div>
```

JavaScript Interface

The JavaScript interface allows you to register handlers on a site-wide level. These are called every time the corresponding event occurs inside the document. The only parameter `obj` is a map containing extended attributes for the event.

```
// Register a listener that is called ... as soon as the plugin is up and running
text20.connector.listener("INITIALIZED", function(obj) {})

// ... a fixation is detected anywhere on the document
// obj.x = document x position
// obj.y = document y position
// obj.type = "START" || "END"
// obj.duration = time in ms the fixation lasted
// obj.meanderivation = how coherent the fixation has been (pixels to center)
text20.connector.listener("fixation", function(obj) {})

// ... reading behavior is detected anywhere in the document
// <Handler is currently being reworked>
text20.connector.listener("perusal", function(obj) {})
```

Advanced Low-Level Configuration

Here we list a set of advanced configuration options for the low-level connector. These have to be set before `core.init()` is being called.

```
// Specify where the plugin-jar is kept
text20.connector.config.archive = "somedir/text20.jar"

// Which eye tracking device to use. Possible choices:
// eyetrackingdevice:auto
// eyetrackingdevice:mouse
// eyetrackingdevice:trackingserver
text20.connector.config.trackingDevice = "eyetrackingdevice:auto"

// URL of the tracking server. 'discover://nearest'
// will do a local search and should work fine in most
// cases.
text20.connector.config.trackingURL = "discover://nearest"

// Enable brain tracking using the Emotiv headset?
text20.connector.config.enableBrainTracker = false

// URL of the brain tracking device
text20.connector.config.brainTrackingURL = "discover://nearest"

// A list of extensions to load (relative path)
text20.connector.config.extensions = ["speechio.jar"]

// If set to true, the plugins will record the
// interaction for later replay and analysis
text20.connector.config.recordingEnabled = false

// If set to true, the plugins will record the
// internal program flow for debugging. Disable to get slight speed ups.
text20.connector.config.diagnosis = true

// If set to true, the plugin will check for updates and report about new versions
// on the console. Also sends an anonymous ID for statistical purposes, but no
// personal information whatsoever.
text20.connector.config.updateCheck =  true

// Where to store our output
text20.connector.config.sessionPath = "/tmp/sessions"
```

```
// Minimum duration in ms for coherent eye tracking data to be considered
// as a fixation
text20.core.config.fixation.minimumDuration = 100

// Maximal radius in pixel for eye tracking data to be considered as
// coherent enough for a fixation
text20.core.config.fixation.maxFixationRadius = 25
```

Using the Java Library Directly

See how to use the Java interface.