

WebExp2 v 1.1 alpha Release Notes: Audio Stimulus, New Features and Bug Fixes

Neil Mayo, nmayo@inf.ed.ac.uk
Informatics, University of Edinburgh
<http://www.webexp.info>

15 March 2006

1 Introduction

This document describes the changes in the newest release of **WebExp2**. These changes distinguish release (version v 1.1 alpha) from the previously distributed release v 1.0 alpha.

2 New Stimulus Components

2.1 Audio Stimuli

It is possible to present an audio file to the subject. This can be done when the slide loads, or you can present a button for the subject to click when they are ready.

To have a clip played when the slide loads, use the **audio** stimulus type:

```
<component>  
  <name>sound</name>  
  <type>audio</type>  
  <content>disappear</content>  
</component>
```

To have a clip played when an associated button is pressed, use the **audiobutton** stimulus type:

```
<component>  
  <name>sound</name>  
  <type>audiobutton</type>  
  <content>appear</content>  
  <label>press me</label>  
</component>
```

The **<label>** parameter is the text that will be displayed on the button.

2.1.1 Preloading audio files

The **<content>** parameter is a reference to a preloaded audio file, which is set up in a parallel manner to the images. In the timeline preamble, declare a **soundbase** where the audio files are to be found,

and a list of `<name>value</name>` pairs describing each sound:

```
<timeline>
  ...
  <soundbase>resources/sounds</soundbase>
  <sounds>
    <appear>KDE_Dialog_Appear.wav</appear>
    <disappear>KDE_Dialog_Disappear.wav</disappear>
  </sounds>
  ...
```

The `name` is referenced in the `content` of an `AudioStimulus`, and the `value` is the file name.

2.2 Selection Response Stimulus

To collect a choice the subject makes from a set of options, you can use the `SelectionResponse` stimulus:

```
<component>
  <name>gender</name>
  <type>selection</type>
  <selType>combo</selType>
  <options>
    <opt01>male</opt01>
    <opt02>female</opt02>
  </options>
</component>
```

The `selType` parameter specifies a particular subtype of selection responses – at the moment the only type is `combo`, which creates a combo-box component with a drop-down list of options. In the near future we will add selection types allowing the options to be presented as radio-buttons (allowing only one choice), and as a list allowing multiple selections.

The options for the selection component should be supplied in an `<options>` element, as the contents of a list of distinctly-named elements. The names of the elements is not important except in ordering the options.

3 TextStimulus text alignment

There is now an `align` property for `TextStimulus` components. The default alignment has been changed from centre to left.

A default alignment for text components can be specified in the slide with the `<align>` parameter. Note that this does NOT affect placement of components within slide - this is specified in the `<rows>` and `<cols>` parameters.

The content of the `<align>` element should be `l`, `c`, or `r` to indicate left, centre or right respectively. You can also use the full words `left`, `centre` (or `center`) or `right`.

4 Partial Results

Partial results are now returned to the server if the subject leaves in the middle of an experiment (by navigating to another page or closing the browser). Partial results are recorded under a subdirectory `partial/` of the experiment directory, with the usual subdirectory structure.

If you want to save partial results to disk for a particular experiment, add the switch `<save-partial-results/>` to the experiment in the `experiments.xml` file. The default is not to save partial results.

5 Improved Feedback Options

5.1 Optional footer feedback

The display of messages in the footer bar of the applet screen is now optional. Currently there is only one message of this sort, indicating that further input is required. Indicators of experiment progress can be disabled in the usual way by designing a stage which does not show progress.

To set the footer messages on or off, include the applet parameter `feedback` in your HTML, with the value `true` or `false`.

```
<param name="feedback" value="true"/>
```

5.2 Optional redirection on experiment completion

It is now possible to redirect the subject to a different URL for a debriefing or summary once the experiment has completed. Just specify the applet parameter `redirect` with a valid URL, and the experiment will automatically redirect to that URL once the final slide has been advanced.

It is advisable to provide a final slide which indicates to the subject that the experiment is over and the redirect is about to occur.

5.3 Optional default completion screen

The screen which was shown at the end of all experiments by default has now been made optional. To use this screen (which now contains only the message “Experiment complete”, add the option element `<completionScreen/>` to the `timeline` in your experiment file.

If you are not using this option or are redirecting to another page after the experiment, you should create your own completion screen as part of the experiment.

6 Bug Fixes

Several bugs, inconsistencies and non-features and were fixed:

- Faulty randomisation algorithm
- Server does not exit when port is unavailable
- Missing field exception on server startup
- Loss of focus when using keyAdvance

- Response output not encoding-aware
- TextResponse input restrictions do not work in Java 1.5

Further details follow:

Faulty randomisation algorithm The implemented algorithm did not randomise correctly for lists of only two items. This is now fixed.

Server The server would provide an error message when the chosen port was in use, but continue to load in an invalid state. It now exits with the error.

The server displayed a `Missing field` exception upon startup when a `maxSubjects` parameter was found in the `experiments.xml` file. This is now fixed.

Loss of focus when using keyAdvance When viewing a slide which uses `keyAdvance`, it was possible to lose the focus of the applet by clicking outside it, and then the slide could not be advanced. Now when the user clicks anywhere in the applet, the default component is focused and it is possible to advance with key presses.

Response output International characters in the subject's responses was not serialised properly to the results files. Now output is written using an encoding (UTF-8 by default).

TextResponse input restrictions Input restrictions in the `TextResponse` component did not work in Java 1.5 due to a slight variation in the `KeyEvent` specification. This is now fixed and restrictions should work in Java 1.4 and above.