MarkWrite 0.1 Brief User Guide

This version...

...is being distributed for beta testing and use. Feel free to pass on to other potential users, but please do not mass-advertise at this stage. Comments, suggestions and bug reports to both Sol Simpson (sol@isolver-software.com) and mark.torrance@ntu.ac.uk.

Preamble

MarkWrite allows markup of handwriting (and, I suppose, drawing) collected via a digitising tablet, Anoto pen, or similar. It does not support capture of these data. It is open source and can be modified and distributed under the terms of the GNU General Public License. We think you'll find it useful, but it is distributed without any warranty and without even the implied warranty of merchantability (whatever that is) or fitness for purpose. Initial programming was by Sol Simpson and funded by School of Social Science, Nottingham Trent University, UK.

Download and requirements

MarkWrite code and standalone executables (32 and 64 bit) can be found here https://github.com/isolver/MarkWrite/releases. It requires Windows 7 or later and may run slowly on older or lower power machines.

Input

Data needs to give time (ms), location (x pixels, y pixels) and pen pressure. Pen pressure can either be dichotomous (pressure > 0 vs. pressure = 0) or continuous. Files may or may not include samples when pressure = 0 (i.e. sampling when the pen is lifted but near the tablet surface). Files may be of one of two sorts:

- Tab-delimited text files with columns for time, x, y, and pressure (Eye and Pen generates these). These must have the extension .txyp to be recognised by MarkWrite and to top row of the file (the column labels) must be exactly T X Y P. See the Test Data folder for examples.

- XML files generated by HandSpy. These must have the extension .xml.

Optionally MarkWrite can also use a taglist file. This is a text file (with extension .txt) that is associated with and has the same name as a specific data file. This contains tag labels for the text, with one label per row. You might, for example, transcribe the text in advance and include all of the words in the text in the taglist. This then (possibly) speeds up the tagging process. The taglist is not essential.

Orientation

The program has several windows, all of which can be resized, moved, and tabified. See image below.

Timeline (top horizontal window) gives separate y and x plots against time. This is used to segment the text into periods for tagging. The blue region bounded by vertical yellow lines is the currently selected segment. The selected segment can be moved and resized using the mouse or by using the shortcut keys described below

Spatial View (central) shows how all or some of the text appeared on the page. The text in green corresponds to the segment that has been selected in the timeline. If the selected segment cannot be tagged (because it is already tagged, or because it cannot be nested within the currently selected parent segment) then it appears in red.

Selected Data (right) shows a spatial view of just the current selection.

Within any of these three windows: Left click and drag moves the view around. Right click and drag zooms. Double clicking in the current selection in the timeline zooms in on this selection.

Segment tree (top left) lists the segment tags that you have created, in the form of a hierarchy. Single left click on a tag label selects that segment associated with that tag. Double clicking zooms to it. Right clicking allows tag editing.

Shortcut keys

In situations where the segments that you want to tag are bounded by pen lifts (e.g., if you want to tag sentences and words) then then the following keys make segment selection must quicker (right and left refer to cursor keys).

right / left	Jump selection forward or backwards
CTRL+left / CTRL+right	Move selection end time forwards or backwards
ALT+left / ALT+right	Move selection start time forwards or backwards
ENTER	Bring up tagging dialogue

Output

File > export (and toolbar buttons) allows creation of both by-sample (large) and by-segment output files. The by-segment file can they be processed in, for example, Excel to give segment duration and the duration of pen-lift ("pause") time prior to segment onset.

Typical coding session

- Open a data file.
- Select the first sentence in the text, either by dragging and expanding the selection with mouse actions or, easier, using shortcut keys until you can see that the whole sentence, and nothing else, is highlighted in the Spatial View (and also appears in the Selected Data window).
- Click the tick in the tool bar or press ENTER and type a name for the tag (e.g., "Sentence 1". Sentence 1 now appears in the Tag Tree.
- Double-click on Sentence 1 in the Tag Tree to zoom in on just this part of the text.
- Select the first word in the sentence. This can be done quickly by dragging the end point of the selection to anywhere in the first word and then CTRL+right.
- Tag this with the name of the first word ("The" perhaps). "The" will appear as a Level 2 tag in the Tag Tree.
- Repeat for all words in the sentence.
- You may (or may not) want to tag letters within each word. Repeat the process above, but this time double click on "The" to make that the focus. Select and label "T". This will appear as a Level 3 tag in the Tag Tree.
- You may (just possibly) want to isolate individual strokes within some or all of the letters. Repeat the process to create stroke tags at Level 4.

- When you are finally finished output both a by-sample data file, and a by-segment data file.

Limitations

- It is not currently possible to save a coding session. This means that you have to complete coding a text within one session (or do some fiddly output-file joining).
- The variables provided by the by-segment output file are currently fairly limited. Some additional calculations, possibly also making use of the by-sample output, will provide what you need. However, we plan in future to add more sophistication to the by-sample output.