

runTrialBlock Instructions

Description

This function takes as input a cell array specifying a block of trials (includes drawing the test stimuli to the screen, playing the audio files, etc) and allows the user to connect the eyetracker, specify trial order, and specify the type of feedback to be given to subjects.

General

This function takes a specified list of trials (pictures to be shown to and sounds to be played for the subject) and runs through them in a specified order.

`trialList` and `windowPtr` must be specified so that the function knows which trial list to display on which screen.

The optional arguments `ET`, `order`, `feedback`, `folder`, and `pictureCoords` can be altered depending on the needs of the experiment.

`ET` turns the eyetracker on,

`order` specifies whether the trials are presented sequentially or randomly,

`feedback` specifies which type of feedback to give the subjects (no feedback for artificial lexicon testing, feedback for artificial lexicon learning, and click until correct most likely for real words),

`folder` specifies where to access the stimuli if they are in a different folder from the one being worked out of, and

`pictureCoords` allows the experimenter to place the image stimuli in any coordinates desired.

This function will return percent correct (`correctness`) from the trials to the main program so that the value can be easily used to evaluate performance.

Usage

```
[correctness] =  
runTrialBlock(trialList,windowPtr,ET,order,feedback,folder,pictureCoords)
```

Ex.

```
runTrialBlock('ExpATrials',5,ETstruct,'RANDOM','FEEDBACK','ExpA/  
,[10 20 30 50; 50 20 70 50; 10 40 30 70; 50 40 70 70]);
```

OR (this is the same function as the one listed above, only using the defined variable names listed below in the **Arguments** section):

```
Ex. runTrialBlock(arg1,arg2,arg3,arg4,arg5,arg6,arg7)
```

Arguments

trialList: String value. The name of the list of trials to be loaded.

Ex. `arg1 = 'ExpATrials'`

windowPtr: Numerical value specifying the window for the stimuli to be printed to.

Ex. `arg2 = 5`

ET: Optionally skip connecting the eyetracker via the numerical input 0. Currently, there is no default for connecting the eyetracker. Either the shell code specifies a variable containing the eyetracker struct, or the user must specify 0 for no eyetracking.

Ex. `arg3 = ETstruct`

order: Optional string value that can specify the trial order to be random via the command 'RANDOM'. Will default to 'SEQUENTIAL'. Inputs are case sensitive, and must be in all caps.

Ex. `arg4 = 'RANDOM'`

feedback: Optional string value that lets the user specify the type of feedback desired. Can choose between 'NOFEEDBACK', 'FEEDBACK', or 'CLICKTILLRIGHT'. Will default to delivering no feedback. Inputs are case sensitive, and must be in all caps.

Ex. `arg5 = 'FEEDBACK'`

folder: String value. Optionally specify the name of the folder containing the stimuli, with a forward slash (/) after the name. It will default to the current folder being worked out of.

Ex. `arg6 = 'ExpA/'`

pictureCoords: Optional numerical values that allow picture coordinates to be specified. Each vector will consist of 4 numbers that denote the locations of the upper left and lower right corners (x,y coordinates) of the image. Can specify the coordinates of multiple pictures by creating new rows for each set of coordinates (maximum number of pictures on screen at a time is 4).

Ex. `arg7 = [10 20 30 50; 50 20 70 50; 10 40 30 70; 50 40 70 70]`

Notes

All arguments that require string values must be entered surrounded by single quotation marks ('').

All inputs are case sensitive.

The (x,y) coordinate system in MATLAB is different from the normal coordinate system.

The (0,0) point is in the upper left corner, and infinity in both directions is in the lower right corner, so the y values must always be larger than the x values.

To specify new rows for the image vectors, use a semi-colon, not the return key.

All vectors must be surrounded by brackets ([]).