msrresearch Documentation

Release 0.1

Björn Guth

July 31, 2015

CONTENTS

1 facereader_client **3**
1.1 Code documentation 3
1.2 Logs 4

Index **5**

Contents:

CHAPTER

ONE

FACEREADER_CLIENT

1.1 Code documentation

class FaceReader.**FaceReader**(*host*, *port*, *buffer_size=4*, *func=None*, *func_args=None*)

FaceReader class to use the API provided by Noldus

The purpose of this class is to operate FaceReader by Noldus. It uses all functionality given by the API documentation from Noldus.

Parameters

- **host** (*str*) – IP address of the host running FaceReader
- **port** (*int*) – Listening port set in FaceReader
- **buffer_size** (*int*) – At the moment I have no idea, why I added this parameter or what it is doing.

class **MySocket**(*host*, *port*, *log=None*, *func=None*, *func_args=None*, *sock=None*)

Class to send and receive messages via TCP.

connect(*host*, *port*)

Connect to a host via TCP using socket.

Parameters

- **host** (*str*) – IP address of host
- **port** (*int*) – Listening port at host

receive()

Receive a message from the set and connected host.

Returns str - the received message

send(*msg*)

Send a message.

Parameters **msg** (*str*) – Message to be send

stop()

Stop the thread in which MySocket is running.

FaceReader.**get_events**()

Receive a list of all event markers set in the current FaceReader project.

Returns A list of all event markers.

FaceReader.**get_stimuli**()

Receive a list of all stimuli set in the current FaceReader project

Returns A list of all stimuli

FaceReader.**score_event**(*event*)
Scores a event marker within the running analysis.

Parameters **event** (*str*) – The name of the event marker to be scored.

FaceReader.**score_stimulus**(*stimulus*)
Scores a stimuli within the running anlysis

Parameters **stimulus** (*str*) – Name of the stimulus to score

FaceReader.**start_analyzing**()
Starts analyzing.

This function sends an action message to FaceReader to start analyzing a preset data source.

FaceReader.**start_detailed_log**()
Starts a log mode, which returns all of the data known to man.

FaceReader.**start_state_log**()
Starts a log mode, which returns the at the moment dominant facial expression.

FaceReader.**stop_analyzing**()
Stops analyzing.

This function sends an action message to Facereader to stop analyzing a preset data source.

FaceReader.**stop_detailed_log**()
Stops the log mode started with *start_detailed_log()*.

FaceReader.**stop_state_log**()
Stops the log mode started with *start_state_log()*.

1.2 Logs

At the moment FaceReader supports to kinds of logs: the detailed and the state log. While the first mentioned includes all data gathered while analyzing a frame, the later log only contains the at the moment dominant facial expression.

1.2.1 Detailed log

Detailed logging can be enabled by calling the function `start_detailed_log()` and can be stopped by calling `stop_detailed_log()`.

1.2.2 State log

State logging can be enabled by calling the function `start_state_log()` and can be stopped by calling `stop_state_log()`.

C

connect() (FaceReader.FaceReader.MySocket method), 3

F

FaceReader (class in FaceReader), 3
FaceReader.MySocket (class in FaceReader), 3

G

get_events() (FaceReader.FaceReader method), 3
get_stimuli() (FaceReader.FaceReader method), 3

R

receive() (FaceReader.FaceReader.MySocket method), 3

S

score_event() (FaceReader.FaceReader method), 3
score_stimulus() (FaceReader.FaceReader method), 4
send() (FaceReader.FaceReader.MySocket method), 3
start_analyzing() (FaceReader.FaceReader method), 4
start_detailed_log() (FaceReader.FaceReader method), 4
start_state_log() (FaceReader.FaceReader method), 4
stop() (FaceReader.FaceReader.MySocket method), 3
stop_analyzing() (FaceReader.FaceReader method), 4
stop_detailed_log() (FaceReader.FaceReader method), 4
stop_state_log() (FaceReader.FaceReader method), 4