/\*\*

@copyright COPYRIGHT 2017 - PROPERTY OF TOBII AB

@copyright 2017 TOBII AB - KARLSROVAGEN 2D, DANDERYD 182 53, SWEDEN - All Rights Reserved.

@copyright NOTICE: All information contained herein is, and remains, the property of Tobii AB and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to Tobii AB and its suppliers and may be covered by U.S.and Foreign Patents, patent applications, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from Tobii AB.

\*/

/\*\*

\* @file tobii\_research\_calibration.h

\* @brief <b>Functionality for implementing calibration.</b>

\*

\*/

#ifndef TOBII\_RESEARCH\_CALIBRATION\_H\_

#define TOBII\_RESEARCH\_CALIBRATION\_H\_

#include "tobii\_research.h"

#ifdef \_\_cplusplus

extern "C" {

#endif

/\*\*

"Defines the overall status of a calibration process.

\*/

typedef enum {

/\*\*

Indicates that the calibration process failed.

\*/

TOBII\_RESEARCH\_CALIBRATION\_FAILURE = 0,

/\*\*

Indicates that the calibration process succeeded.

\*/

TOBII\_RESEARCH\_CALIBRATION\_SUCCESS = 1

} TobiiResearchCalibrationStatus;

/\*\*

Defines the validity of calibration eye data.

\*/

typedef enum {

/\*\*

The eye tracking failed or the calibration eye data is invalid.

\*/

TOBII\_RESEARCH\_CALIBRATION\_EYE\_VALIDITY\_INVALID\_AND\_NOT\_USED = -1,

/\*\*

Eye Tracking was successful, but the calibration eye data was not used in calibration e.g. gaze was too far away.

\*/

TOBII\_RESEARCH\_CALIBRATION\_EYE\_VALIDITY\_VALID\_BUT\_NOT\_USED = 0,

/\*\*

The calibration eye data was valid and used in calibration.

\*/

TOBII\_RESEARCH\_CALIBRATION\_EYE\_VALIDITY\_VALID\_AND\_USED = 1,

/\*\*

The calibration eye data has an unexpected validity.

\*/

TOBII\_RESEARCH\_CALIBRATION\_EYE\_VALIDITY\_UNKNOWN

} TobiiResearchCalibrationEyeValidity;

/\*\*

Represents the calibration sample data collected for one eye.

\*/

typedef struct {

/\*\*

The eye sample position on the active display Area for the left eye.

\*/

TobiiResearchNormalizedPoint2D position\_on\_display\_area;

/\*\*

Information about if the sample was used or not in the calculation for the left eye.

\*/

TobiiResearchCalibrationEyeValidity validity;

} TobiiResearchCalibrationEyeData;

/\*\*

Represents the data collected for a calibration sample.

\*/

typedef struct {

/\*\*

The calibration sample data for the left eye.

\*/

TobiiResearchCalibrationEyeData left\_eye;

/\*\*

The calibration sample data for the right eye.

\*/

TobiiResearchCalibrationEyeData right\_eye;

} TobiiResearchCalibrationSample;

/\*\*

Represents the Calibration Point and its collected calibration samples.

\*/

typedef struct {

/\*\*

The position of the calibration point in the Active Display Area.

\*/

TobiiResearchNormalizedPoint2D position\_on\_display\_area;

/\*\*

An array of calibration samples.

\*/

TobiiResearchCalibrationSample\* calibration\_samples;

/\*\*

Number of calibration calibration points in the array.

\*/

size\_t calibration\_sample\_count;

} TobiiResearchCalibrationPoint;

/\*\*

Represents the result of the calculated calibration.

\*/

typedef struct {

/\*\*

Array of calibration points.

\*/

TobiiResearchCalibrationPoint\* calibration\_points;

/\*\*

Number of calibration calibration points in the array.

\*/

size\_t calibration\_point\_count;

/\*\*

Gets the status of the calculation.

\*/

TobiiResearchCalibrationStatus status;

} TobiiResearchCalibrationResult;

/\*\*

Represents the result of the calculated HMD based calibration.

\*/

typedef struct {

/\*\*

Gets the status of the calculation.

\*/

TobiiResearchCalibrationStatus status;

} TobiiResearchHMDCalibrationResult;

/\*\*

@brief Enters the screen based calibration mode and the eye tracker is made ready for collecting data and calculating new calibrations.

\snippet calibration.c CalibrationEnterExample

@param eyetracker: Eye tracker object.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL

tobii\_research\_screen\_based\_calibration\_enter\_calibration\_mode(

TobiiResearchEyeTracker\* eyetracker);

/\*\*

@brief Leaves the screen based calibration mode.

\snippet calibration.c CalibrationLeftExample

@param eyetracker: Eye tracker object.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL

tobii\_research\_screen\_based\_calibration\_leave\_calibration\_mode(

TobiiResearchEyeTracker\* eyetracker);

/\*\*

@brief Starts collecting data for a calibration point.

The argument used is the point the calibration user is assumed to

be looking at and is given in the active display area coordinate system.

You must call tobii\_research\_calibration\_enter\_calibration\_mode before calling this function.

This function is blocking while collecting data and may take up to 10 seconds.

\snippet calibration.c CalibrationExample

@param eyetracker: Eye tracker object.

@param x: Normalized x coordinate on active display area where the user is looking.

@param y: Normalized y coordinate on active display area where the user is looking.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL tobii\_research\_screen\_based\_calibration\_collect\_data(

TobiiResearchEyeTracker\* eyetracker,

float x,

float y);

/\*\*

@brief Removes the collected data associated with a specific calibration point.

\snippet hmd\_calibration.c ReCalibrationExample

@param eyetracker: Eye tracker object.

@param x: Normalized x coordinate of point to discard.

@param y: Normalized y coordinate of point to discard.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL tobii\_research\_screen\_based\_calibration\_discard\_data(

TobiiResearchEyeTracker\* eyetracker,

float x,

float y);

/\*\*

@brief Uses the collected data and tries to compute calibration parameters.

If the calculation is successful, the result is applied to the eye tracker.

If there is insufficient data to compute a new calibration or if the collected

data is not good enough then calibration is failed and will not be applied.

\snippet hmd\_calibration.c CalibrationExample

@param eyetracker: Eye tracker object.

@param result: Represents the result of the calculated calibration.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL tobii\_research\_screen\_based\_calibration\_compute\_and\_apply(

TobiiResearchEyeTracker\* eyetracker,

TobiiResearchCalibrationResult\*\* result);

/\*\*

@brief Free memory allocation for the calibration result received via @ref tobii\_research\_screen\_based\_calibration\_compute\_and\_apply.

@param result: Calibration result to free.

\*/

TOBII\_RESEARCH\_API void TOBII\_RESEARCH\_CALL tobii\_research\_free\_screen\_based\_calibration\_result(

TobiiResearchCalibrationResult\* result);

/\*\*

@brief Enters the hmd based calibration mode and the eye tracker is made ready for collecting data and calculating new calibrations.

\snippet hmd\_calibration.c HMDCalibrationEnterExample

@param eyetracker: Eye tracker object.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL

tobii\_research\_hmd\_based\_calibration\_enter\_calibration\_mode(

TobiiResearchEyeTracker\* eyetracker);

/\*\*

@brief Leaves the hmd based calibration mode.

\snippet hmd\_calibration.c HMDCalibrationLeftExample

@param eyetracker: Eye tracker object.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL

tobii\_research\_hmd\_based\_calibration\_leave\_calibration\_mode(

TobiiResearchEyeTracker\* eyetracker);

/\*\*

@brief Starts collecting data for a calibration point.

The argument used is the point the calibration user is assumed to be looking at and is given in the HMD coordinate system.

You must call tobii\_research\_screen\_based\_calibration\_enter\_calibration\_mode before calling this function.

This function is blocking while collecting data and may take up to 10 seconds.

@param eyetracker: Eye tracker object.

@param x: x coordinate in the HMD coordinate system where the user is looking.

@param y: y coordinate in the HMD coordinate system where the user is looking.

@param z: z coordinate in the HMD coordinate system where the user is looking.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL tobii\_research\_hmd\_based\_calibration\_collect\_data(

TobiiResearchEyeTracker\* eyetracker, float x, float y, float z);

/\*\*

@brief Uses the collected data and tries to compute calibration parameters.

If the calculation is successful, the result is applied to the eye tracker.

If there is insufficient data to compute a new calibration or if the collected

data is not good enough then calibration is failed and will not be applied.

\snippet hmd\_calibration.c HMDCalibrationExample

@param eyetracker: Eye tracker object.

@param result: Represents the result of the calculated HMD calibration.

@returns A @ref TobiiResearchStatus code.

\*/

TOBII\_RESEARCH\_API TobiiResearchStatus TOBII\_RESEARCH\_CALL tobii\_research\_hmd\_based\_calibration\_compute\_and\_apply(

TobiiResearchEyeTracker\* eyetracker,

TobiiResearchHMDCalibrationResult\* result);

#ifdef \_\_cplusplus

}

#endif

#endif /\* TOBII\_RESEARCH\_CALIBRATION\_H\_ \*/