UML Diagram Team 57

HeartWaveControl

+ getSetChallengeLvl() : QString

+ getChallNum(): int

+ getSetBreath(): int

+ setBreath(int): void

+setChallenge(int): void

+ achScore(int,int): int

+ challengeLvl(int,float): QString

+ coherenceScore(float,int): int

+ achScore(int,int): int

+ challengeLvl(int,float): QString

Coherence

Settings

- breathPacer: int

- challengeLvl: int

- levels: QStringList

+ getChallengeLvl(): QString

+ getChallNum(): int

+ getBreathPacer(): int

+ setChallengeLvl(int): void

+ setBreathPacer(int): void

1

1

1

1

Used to create the graph.

QCustomPlot

LogHistory

+ loadLogs(const QJsonArray &): void

+ getLogs(): const QVector<LogEntry>

+ getTimestamps(): QVesctor<QDateTime>

+ addLogEntry(const LogEntry&): void

+ resetLog(): void

+ deleteLog(int): void

level: QString

time: int

struct: CoherenceData

struct: LogEntry

time: QDateTime

length: int

average\_coherence: double

achievement\_score: int

plotXData: QVector<double>

plotYData: QVector<double>

1

0…\*

1

0…\*

MainWindow

1

1

1

1

LogHistory

HeartWaveControl

MainWindow

1

1

1

1

- dataTimer: QTimer

- powerLevel: int

- power: bool

- sensorAttached: bool

- mainMenu: bool

- mainMenuItems: QStringList

- settings: bool

- settingItems: QStringList

- logHistorySelected: bool

- logSelected: int

- logDetailsSelected: bool

- coherence: bool

- setBreath: bool

- setChallenge: bool

- setChallenges: QStringList

- resetDevice: bool

- connectButtons(): void

- powerOnOff(): void

- setSelected(): void

- setupPlot(QCustomPlot \*): void

- setupStaticLogPlot(QCustomPlot\*, const QVector<double>, const QVector<double> , int):void

- getPlotData(QCustomPlot\* , QVector<double>, QVector<double>): void

- displayLogDetails(const LogEntry&): void

- coherenceTest(float): void

- sessionEnd(): void

- showSessionSummary(): void

- logSession(): void

- makeMenu(): void

- setMenu(): void

- losePower(int): void

- breathPace(int): void

- lights(): void

- sensorFunction(): void

- sensorTracker:bool

- redLight: bool

- blueLight: bool

- greenLight: bool

- highScoreCheck: bool

- mediumScoreCheck: bool

- lowScoreCheck: bool

- timeHighSec: int

- timeMediumSec: int

- timeLowSec: int

- lowScore: QList<float>

- mediumSore: QList<float>

- highScore: QList<float>

- lIt: QList<float>::iterator

- mIt: QList<float>::iterator

- hIt: QList<float>::iterator