**Wearable Epilepsy Device & MC Team**

**T4: Patterns Matching, Association and Prediction**

# Signal.h

* Overview
  + Declares the variables and function signatures used in the Signal.cpp.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Variables
  + frequency
    - Integer value that represents the incoming frequency.
* Function Signatures
  + Signal();
  + Signal(int);
  + ~Signal();
  + void setFrequency(int);
  + int getFrequency();

# Signal.cpp

* Overview
  + The Signal Class is used to create a Signal object that monitors signals being received. The Signal Object’s Integer value will be constantly changed to represent if the Signals frequency.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Signal()
    - Default Constructor of the Signal. Initializes the frequency to be 0.
  + Signal(int frequency)
    - Constructor for when a value has been designated during Signal object creation. Sets the frequency.
    - @param frequency - Integer value to set frequency with.
  + ~Signal()
    - Default Destructor that deletes the Signal. Prevents Memory Leak.
  + setFrequency(int frequency)
    - Sets the frequency variable with an Integer value.
    - @param frequency - Integer value to set frequency with.
  + getFrequency()
    - Returns the frequency of the Signal.
    - @return frequency - Integer value for the frequency.

# Issue.h

* Overview
  + Declares the variables and function signatures used in the Issue.cpp.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Variables
  + message
    - String value that displays the Issue.
  + pattern[5]
    - Boolean array that represents the Issue Pattern.
* Function Signatures
  + Issue();
  + Issue(string message, bool pattern[]);
  + ~Issue();
  + void setMessage(string message);
  + void setPattern(bool pattern[]);
  + bool\* getPattern();
  + string getMessage();

# Issue.cpp

* Overview:
  + Issue Class is used to create an Issue object that represents an Issue Pattern. The Issue Pattern will be compared with signals to see if that specific Issue is occurring.
* Authors:
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Issue()
    - Default Constructor of the Issue. Initializes the pattern to be all true. Initializes the message as 'Detected an Issue'.
  + Issue(string message, bool pattern[]);
    - Constructor for when values have been designated during Issue object creation.Sets the message for the Issue.Sets the pattern for the Issue.
    - @param message - String to set the message with.
    - @param pattern[] - Boolean array to set the pattern with.
  + ~ Issue()
    - Default Destructor that deletes the Issue. Prevents Memory Leak.
  + setMessage(string message)
    - Sets the message variable with a String value.
    - @param message - String to set the message with.
  + setPattern(bool pattern[])
    - Sets the pattern array.
    - @param pattern - The array to set the pattern array with.
  + getPattern()
    - Returns Issue Pattern.
    - @return bool\* - Boolean array to set the pattern with.
  + getMessage()
    - Returns the Issue message.
    - @return string - String containing the Issue message.

# Patient.h

* Overview
  + Declares the variables and function signatures used in the Patient.cpp.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Variables
  + stats[2][5]
    - 2D Integer Array that represents the normal ranges for the patient.
  + isAbnormal [5]
    - Boolean Array that represents the pattern of abnormal signals.
* Function Signatures
  + Patient();
  + Patient(int stats[][5]);
  + ~Patient();
  + void setStats(int stats[][5]);
  + int getStats(int, int);
  + void checkRanges(Signal signals[]);
  + string matchPattern(Issue issues[]);

# Patient.cpp

* Overview
  + Patient Class is used to create an Patient object that represents a Patient. The Patient will store the normal ranges for each signal as stats. These stats are compared to the signal values and determines if the signals or abnormal or not. Abnormalities are stored in an array. The abnormal array is then compared with Issues to see if the abnormal signals match an Issue pattern that denotes a seizure.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Patient ()
    - Default Constructor of the Patient. Initializes the Patient stats to be all 0. Initializes the isAbnormal array to be all false.
  + Patient(int stats[][5])
    - Constructor for when attibutes are also passed in during the Patient's creation. Sets the Patient stats array. Sets the isAbnormal array all to false.
    - @param stats - The 2D Intege Array of provided stats.
  + ~Signal()
    - Default Destructor that deletes the Patient. Prevents Memory Leak.
  + setStats(int stats[][5])
    - Sets the Patient stats array with the provided stats.
    - @param stats - The array of provided stats.
  + getStats(int row, int col)
    - Returns the specified stat at a certain row and column.
    - @param row - The index of the desired row.
    - @param col - The index of the desired column.
    - @return int - Integer containing the specified stat.
  + checkRanges(Signal signals[])
    - Checks to see if the Signal frequencies are within the Patient ranges. Sets corresponding Boolean values within the isAbnormal array.
    - @param signals - The Signal array to be compared with.
  + matchPattern(Issue issues[])
    - Compares the Signals in the Signal array with the isAbnormal pattern.
    - @param issues - The Issue array to be compared with.
    - @returns – Returns message String representing Issue status.

# Online Code Link

* The code and project artifacts were uploaded to GitHub for easy access.
  + <https://github.com/Terrillc13/PatternsMatchingAssociationPrediction/tree/OriginalProject>
* The code and project artifacts will also be included in the .zip file.
  + - ssue is occurring.