**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
  + isAbnormal
    - Boolean value that shows if signal is abnormal or not.
* Function Signatures
  + Signal();
  + Signal(bool);
  + ~Signal();
  + void setIsAbnormal(bool);
  + bool getIsAbnormal();

# Signal.cpp

* Overview
  + Signal Class is used to create a Signal object that monitors signals being recieved. The Signal object's isAbnormal Boolean value will be constantly changed to represent if the Signals are producing an abnormality or not.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Signal()
    - Default Constructor of the Signal. Initializes isAbnormal to be false.
  + Signal(bool isAbnormalTemp)
    - Constructor for when a value has been designated during Signal object creation. Sets the isAbnormal.
    - @param isAbnormalTemp - Boolean value to set isAbnormal with.
  + ~Signal()
    - Default Destructor that deletes the Signal. Prevents Memory Leak.
  + setIsAbnormal(bool isAbnormalTemp)
    - Sets the isAbnormal variable with a boolean value.
    - @param isAbnormalTemp - Boolean value to set isAbnormal with.
  + getIsAbnormal ()
    - Returns whether a signal is abnormal or not.
    - @return isAbnormal - Boolean value that shows if signal is abnormal or not.

# 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
    - Boolean array that represents the Issue Pattern.
  + Function Signatures
  + Issue();
  + Issue(bool\*, string);
  + ~Issue();
  + void setPattern (bool\*);
  + void setMessage(string);
  + 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 to see if that specific Issue is occurring.
* Authors:
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Issue()
    - Default Constructor of the Issue. Initialize the size of the boolean array to 5. Initializes the pattern to be all true. Initializes the message as 'Detected an Issue'.
  + Issue(bool\* patternTemp, string messageTemp)
    - Constructor for when a boolean array and message has been designated during the Issue object creation. Sets the pattern array. Sets the message.
    - @param patternTemp - The array to set the pattern array with.
    - @param messageTemp - String value that contains the Issue.
  + ~ Issue()
    - Default Destructor that deletes the Issue. Prevents Memory Leak.
  + setPattern(bool\* patternTemp)
    - Sets the pattern array.
    - @param patternTemp - The array to set the pattern array with.
  + setMessage(string messageTemp)
    - Sets the message variable with a String value.
    - @param messageTemp - String value that contains the Issue.
  + getPattern()
    - Returns Issue Pattern.
    - @return bool\* - Pointer to the Boolean pattern array.
  + getMessage()
    - Returns the Issue message.
    - @return string - String containing the Issue message.

# Seizure.h

* Overview
  + Declares the variables and function signatures used in the Seizure.cpp.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Variables
  + NONE
* Function Signatures
  + Seizure();
  + ~Seizure();
  + String checkForSeizure(Signal\*, Issue\*);

# Seizure.cpp

* Overview
  + Seizure Class is used as a control class that compares the Signal objects with the Issue object patterns and determines if there is a problem being outlined by the signals.
* Authors
  + Clayton D. Terrill and Ian S. Barney
* Methods:
  + Seizure()
    - Default Constructor of the Issue. Only needed to access the class.
  + ~ Seizure()
    - Default Destructor that deletes the Seizure class. Prevents Memory Leak. No memory allocated in this class, so no deallocation is needed.
  + checkForSeizure(Signal\* signalArray, Issue\* issueArray)
    - Compares the Signals in the Signal array with each Issue's pattern.
    - @param signalArray - Pointer to the signalArray which allows access.
    - @param issueArray - Pointer to the issueArray which allows access.
    - @return bool\* - Pointer to the Boolean pattern array.

# Online Code Link

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