ModelMain

Vocabulary

-currentVocabulary
-currentQuestionIndex
-string currentQuestion
-string CurrentAnswer
-string fName
-vector questions
-vector answers
-vector fileNames
-vector askedQuestions
-bool vocabularyLoaded

+Vocabulary()
+vector getAskedQuestions()
+void returnRandomQuestionFromVocabulary()
+int randomInt(int start, int end)
+int returnQuestionIndex()
+string returnCurrentQuestion()
+string returnCurrentAnswer()
+string accessQuestionViaIndex(int i)
+string accessAnswerViaIndex(int i)
+bool isInVector(vector v, int i)
+void addAskedQuestions(int i)
+void loadVocabularyFile()

QuestionAlgorithm

-int lineCount
-int lineNoPlayerName
-int sizeOfCurrentBracket;
-int currentQuestionIndex;
-string cfgLine;
-const string fileName;
-string playerName;
-string cfgElement;
-vector tmpVector;
-vector b1
-vector b2
-vector b3
-vector b4
-bool newPlayer
-vector *b1pt
-vector *b2pt
-vector *b3pt
-vector *b4pt

-bool isNewPlayer()
-bool hasConfig()
-int renturnRandomDelete(vector v)
-void segmentCfg()
-void splitCfg()
-int returnConfigLine(ifstream& f)
-void setCurrentQuestionIndex(int i)
-void delete_line(const char* file_name, int n)
+QuestionAlgorithm(string n)
+void loadCfg()
+void addResultToCfgElement(int index, float performance)
+int returnRandomQuestion()
+int returnCurrentQuestionIndex()
+void setPlayerName(string n)
+void writeToConfig()
+string returnCfg()
+string returnCfgElement()

Game

-int gameTime
-int gameRounds
-int currentRound
-string pName
-string welcomeMessage 1
-bool gameRunning

+Game(int agrc, char* argc[])
+void endGame()
+void delayBy(float f, Timer* t, bool b)
+void clearScreen()
+void checkArg(int argc, char* argv[])
+void presentWelcomeMessage()
+void setGameTime(int i)
+void incrementRoundCount()
+void setGameRounds(int i)
+void setPlayerName(string n)
+int getGameTime()
+int getGameRounds()
+int getCurrentRound()
+string getPlayerName()

Timer

-bool timerRunning
-timerStartTime
-timerEndTime

+void start()
+void stop()
-double elapsedMS()
+double elapsedSecs()

InputCompare

-string answer
-string userInput
-string *answerptr
-string *userInputptr
+int maxScore
+float scorePercentage
+Attribute3
+Attribute4

+float fuzzyComparison()
+int determineHealthPointsToLoose(float p)
-string toLower(string str)
-string removeWhitespace(string str)
-void addSpaces(string *original, string *in)
+void setAnswer(string n)
+void setInput(string n)