Text2SpeechEditor

Sprint No 2 Report

Team Name: Φραπές και Ξενύχτι
ΑΝΔΡΟΥΤΣΟΣ ΙΩΑΝΝΗΣ, ΑΜ:3180
ΠΙΠΙΔΗΣ ΠΑΣΧΑΛΗΣ, ΑΜ: 3064

VERSIONS HISTORY

Date	Version	Description	Author
24/5/2020	V1.0	First complete version	ΠΙΠΙΔΗΣ ΠΑΣΧΑΛΗΣ

1 Introduction

This document provides information concerning the **2nd** sprint of the project.

1.1 Purpose

The purpose of this report is to include information about the software developed during the project's course as well as it's tests and design.

1.2 Document Structure

The rest of this document is structured as follows. Section 2 describes out Scrum team and specifies the this Sprint's backlog. Section 3 specifies the main design concepts for this release of the project.

2 Scrum team and Sprint Backlog

2.1 Scrum team

Product Owner	Apostolos Zarras
Scrum Master	Pipidis Paschalis
Development Team	Androutsos Ioannis, Pipidis Paschalis

2.2 Sprint Backlog

All user stories have been realized during this sprint. More specifically:

[US-1] As a user, I want to create a new empty document, by giving its title and author. The application should automatically record the creation date.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Check if a new document is created successfully.	Title: Just a test Author: Java Robot	A new document with title “Just a test” and author “Java Robot” should be created	Document created successfully with the expected title and author.	Pass

[US-2] As a user, I want to edit the contents of the document, via the application's user interface.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
2	Check if internal document edits were saved successfully.	Text edit: Just robot things	The internal document should be updated with the text “Just robot things”	Document updated successfully with the expected text.	Pass

[US-3] As a user, I want to save the contents of the document to disk by providing a particular filename. The application should automatically record the save date.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
3	Check if the current document data were saved to a file on the disk successfully.	File name: RobotTestFile.txt File text: Just robot things	A new file named “RobotTestFile.txt” should be created and it should contain the text “Just robot things”	File created successfully with the expected text.	Pass

[US-4] As a user, I want to open the contents of an existing document from disk by providing a particular file path, or by browsing the file system folders.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
4	Check if the current document data were successfully overwritten by the data in a file on the disk.	File name: RobotTestFile.txt File text: Just robot things	The data from a file named "RobotTestFile.txt" should be loaded and stored in the internal document	File opened and loaded successfully with the expected text stored in the program's internal document.	Pass

[US-5] As a user, I want to transform the contents of the document to speech.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
5	Check if the entire current text from the text editor was transformed to speech successfully.	Editor text: Just robot things	The text from the text editor "Just robot things" should be loaded into the internal document and transformed to speech.	Text loaded and transformed successfully.	Pass

[US-6] As a user, I want to select a line and transform it to speech.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
6	Check if a line of text from the text editor was transformed to speech successfully.	Editor text: Just robot things Line transformed: Line number 1	The text from the 1st line of the text editor "Just robot things" should be loaded into the internal document and transformed to speech.	Text from line 1 of the editor loaded and transformed successfully.	Pass

[US-7] As a user I want to transform the contents of the document to speech in reverse, i.e. play the last word of the last line first and so on.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
7	Check if the entire current text from the text editor was reversed and transformed to speech successfully.	Editor text: Just robot things Reverse text: things robot Just	The text from the text editor “Just robot things” should be loaded into the internal document, reversed and transformed to speech.	Text loaded, reversed and transformed successfully.	Pass

[US-8] As a user I want to select a line and transform it to speech in reverse, i.e. and play the last word first and so on.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
8	Check if a line of text from the text editor was reversed and transformed to speech successfully.	Editor text: Just robot things Line transformed: Line number 1	The text from the 1st line of the text editor “Just robot things” should be loaded into the internal document, reversed and transformed to speech.	Text from line 1 of the editor loaded, reversed and transformed successfully.	Pass

[US-9] As a user I want to encode the contents of the document and then transform them to speech.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
9	Check if the entire current text from the text editor was encoded and transformed to speech successfully.	Editor text: Just robot things Encoded text: Qfhg ilylg gsrmth	The text from the text editor “Just robot things” should be loaded into the internal document, encoded and transformed to speech.	Text loaded, encoded and transformed successfully.	Pass

[US-10] As a user I want to select a line, encode it and transform it to speech.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
10	Check if a line of text from the text editor was encoded and transformed to speech successfully.	Editor text: Just robot things Encoded text: Qfhg ilylg gsrmth Line : #1	The text from the 1st line of the text editor "Just robot things" should be loaded into the internal document, encoded and transformed to speech.	Text from line 1 of the editor loaded, encoded and transformed successfully.	Pass

[US-11] As a user I want to tune the text encoding technique. In particular the application should support at least the following encoding strategies:

- Atbash: The Atbash cipher is formed by taking the alphabet and mapping it to its reverse, so that the first letter becomes the last letter, the second letter becomes the second to last letter, and so on.
- Rot-13: Rot-13 is a letter substitution cipher that replaces a letter with the 13th letter after it, in the alphabet. Rot-13 is a special case of the Caesar cipher, which was developed in ancient Rome.

Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
11	Check if the default encoding technique of Atbash was switched to ROT13 successfully.	Default technique: Atbash Technique chosen: ROT13	The encoding technique should change to ROT13	Encoding technique switched to ROT13 successfully	Pass

[US-12] As a user I want to be able to tune the audio parameters, i.e., the volume, the speech rate and the pitch.

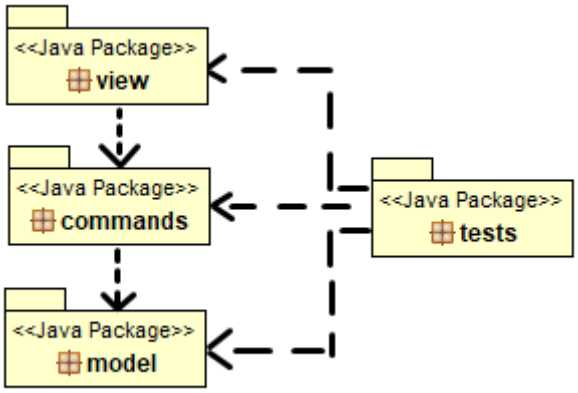
Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
12	Check if the audio parameters of volume, voice pitch and speech rate were changed successfully.	Initial values: Volume=50, Pitch=100, Rate=150. New values: Volume=25, Pitch=75, Rate=100.	The values of volume, pitch and rate should change to 25, 75 and 100 respectively.	The values of volume, pitch and rate successfully changed to 25, 75 and 100 respectively.	Pass

[US-13] As a user I want to be able to store a sequence of actions/commands (e.g. open file, edit contents, play contents, play line, save file) that I have performed in main memory and re-execute them multiple times.

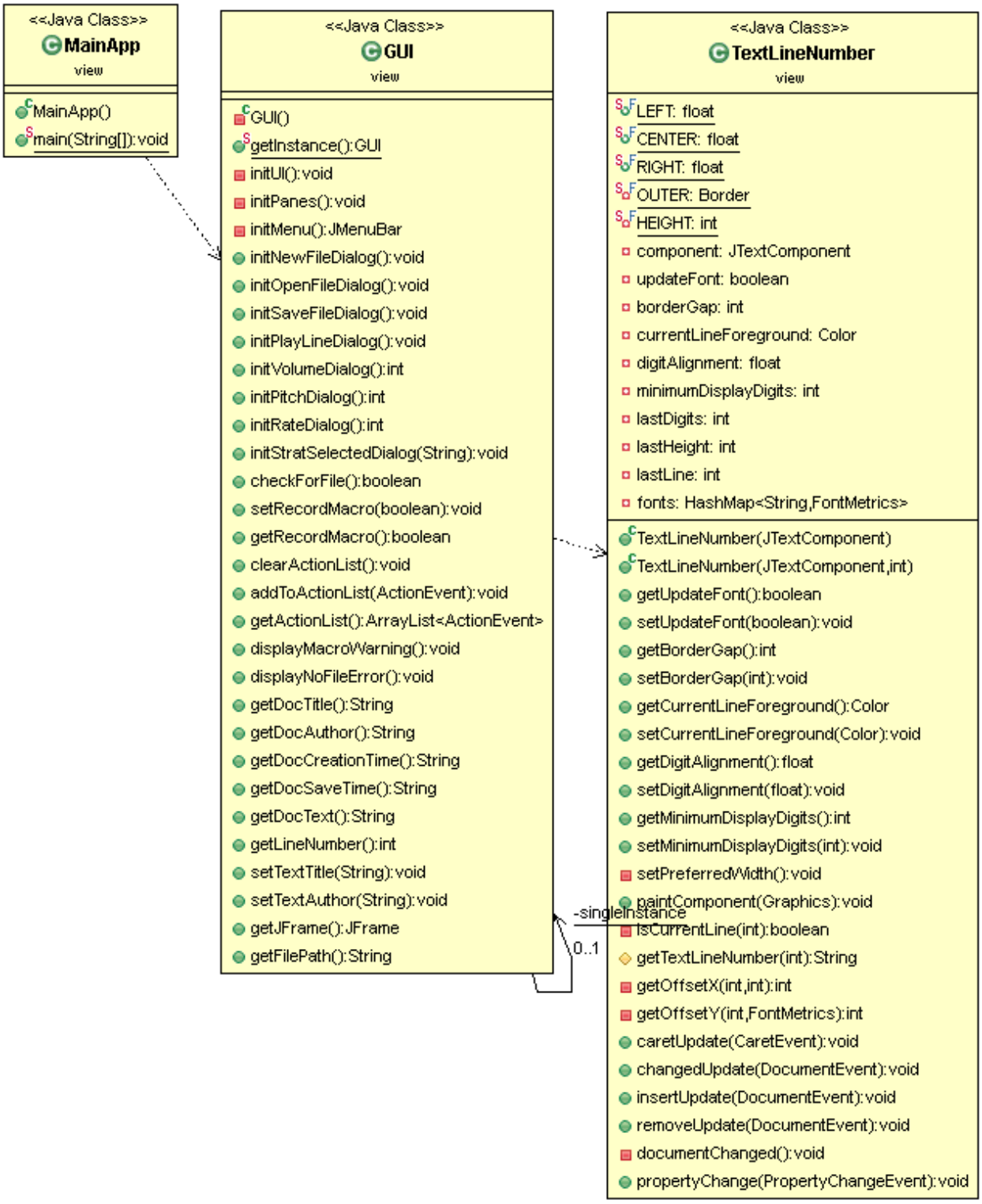
Test Case #	Description	Test Data	Expected Result	Actual Result	Pass/ Fail
13	Check if a list of commands the user indicates the start and end of has been recorded and replayed successfully.	Command list:New File with title “Just a test macro” and author “Java Robot”	A new file with the title “Just a test macro” and author “Java Robot” should be created when the “Use recorded macro” button is pressed	Command re-executed successfully and a new file with the correct data was created.	Pass

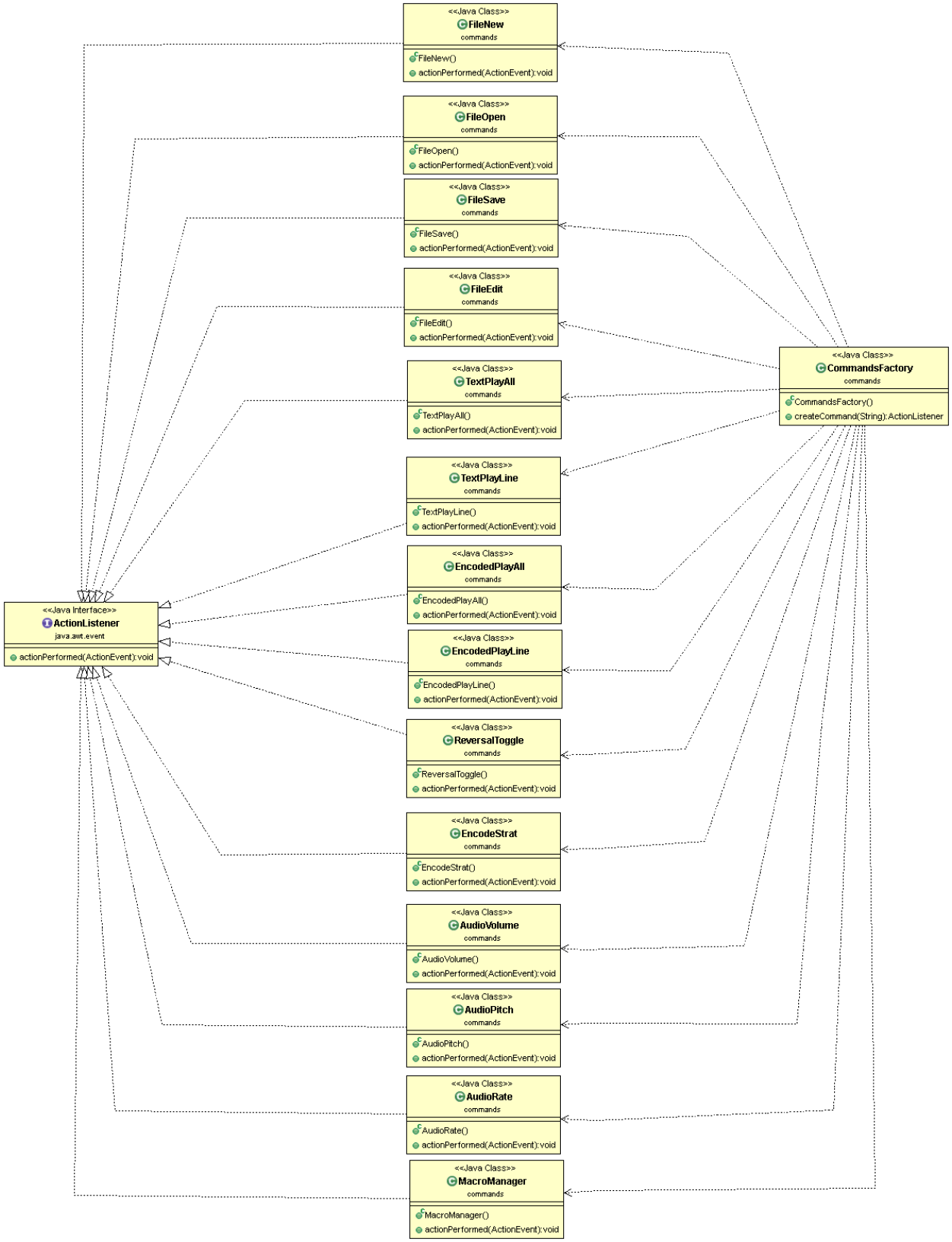
3 Design

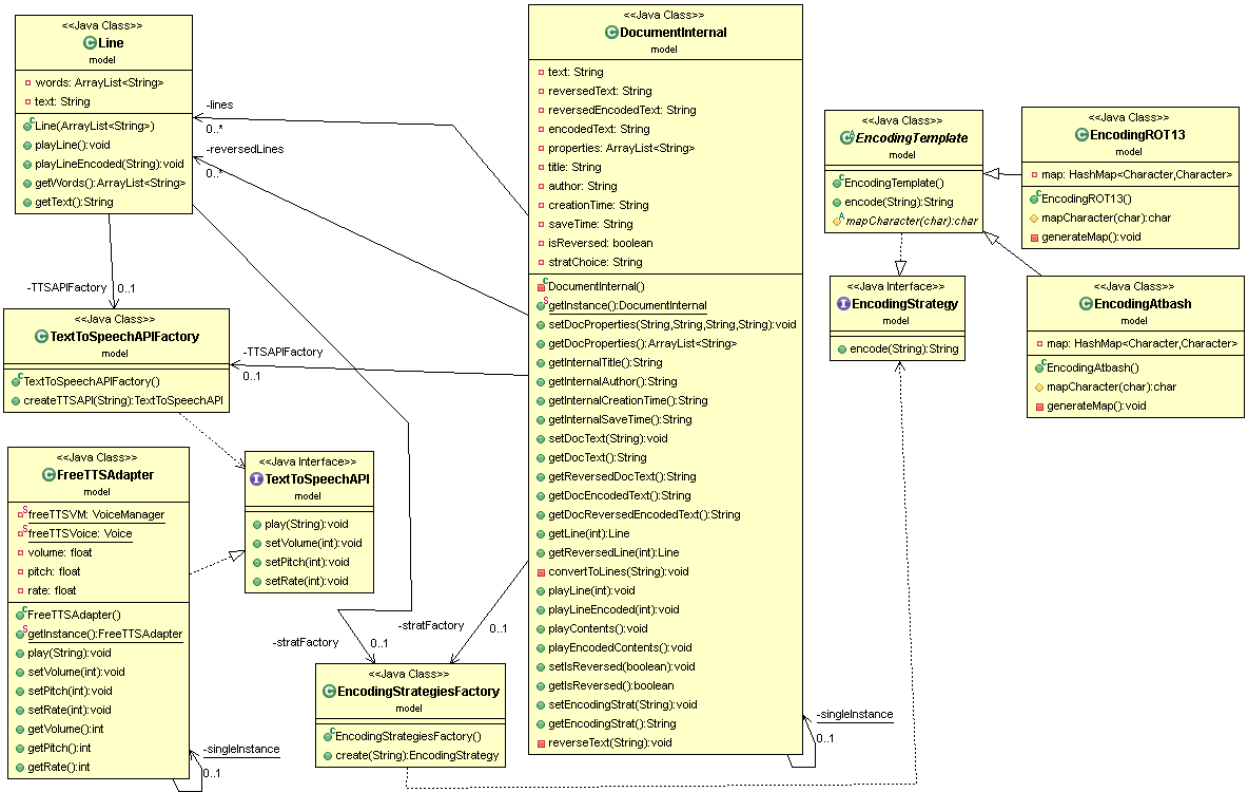
3.1 Architecture



3.2 Design


<<Java Class>>
MainApp
view
MainApp()
main(String[]):void
<<Java Class>>
GUI
view
GUI()
getInstance():GUI
initUI():void
initPanes():void
initMenu():JMenuBar
initNewFileDialog():void
initOpenFileDialog():void
initSaveFileDialog():void
initPlayLineDialog():void
initVolumeDialog():int
initPitchDialog():int
initRateDialog():int
initStratSelectedDialog(String):void
checkForFile():boolean
setRecordMacro(boolean):void
getRecordMacro():boolean
clearActionList():void
addToActionList(ActionEvent):void
getActionList():ArrayList<ActionEvent>
displayMacroWarning():void
displayNoFileError():void
getDocTitle():String
getDocAuthor():String
getDocCreationTime():String
getDocSaveTime():String
getDocText():String
getLineNumber():int
setTextTitle(String):void
setTextAuthor(String):void
getJFrame():JFrame
getFilePath():String
<<Java Class>>
TextLineNumber
view
LEFT: float
CENTER: float
RIGHT: float
OUTER: Border
HEIGHT: int
component: JTextComponent
updateFont: boolean
borderGap: int
currentLineForeground: Color
digitAlignment: float
minimumDisplayDigits: int
lastDigits: int
lastHeight: int
lastLine: int
fonts: HashMap<String,FontMetrics>
TextLineNumber(JTextComponent)
TextLineNumber(JTextComponent,int)
getUpdateFont():boolean
setUpdateFont(boolean):void
getBorderGap():int
setBorderGap(int):void
getCurrentLineForeground():Color
setCurrentLineForeground(Color):void
getDigitAlignment():float
setDigitAlignment(float):void
getMinimumDisplayDigits():int
setMinimumDisplayDigits(int):void
setPreferredWidth():void
paintComponent(Graphics):void
isCurrentLine(int):boolean
getTextLineNumber(int):String
getOffsetX(int,int):int
getOffsetY(int,FontMetrics):int
caretUpdate(CaretEvent):void
changedUpdate(DocumentEvent):void
insertUpdate(DocumentEvent):void
removeUpdate(DocumentEvent):void
documentChanged():void
propertyChange(PropertyChangeEvent):void
-singleInstance
0..1


<<Java Class>>
FileNew
commands
FileNew()
actionPerformed(ActionEvent):void
<<Java Class>>
FileOpen
commands
FileOpen()
actionPerformed(ActionEvent):void
<<Java Class>>
FileSave
commands
FileSave()
actionPerformed(ActionEvent):void
<<Java Class>>
FileEdit
commands
FileEdit()
actionPerformed(ActionEvent):void
<<Java Class>>
TextPlayAll
commands
TextPlayAll()
actionPerformed(ActionEvent):void
<<Java Class>>
TextPlayLine
commands
TextPlayLine()
actionPerformed(ActionEvent):void
<<Java Class>>
EncodedPlayAll
commands
EncodedPlayAll()
actionPerformed(ActionEvent):void
<<Java Class>>
EncodedPlayLine
commands
EncodedPlayLine()
actionPerformed(ActionEvent):void
<<Java Class>>
ReversalToggle
commands
ReversalToggle()
actionPerformed(ActionEvent):void
<<Java Class>>
EncodeStrat
commands
EncodeStrat()
actionPerformed(ActionEvent):void
<<Java Class>>
AudioVolume
commands
AudioVolume()
actionPerformed(ActionEvent):void
<<Java Class>>
AudioPitch
commands
AudioPitch()
actionPerformed(ActionEvent):void
<<Java Class>>
AudioRate
commands
AudioRate()
actionPerformed(ActionEvent):void
<<Java Class>>
MacroManager
commands
MacroManager()
actionPerformed(ActionEvent):void
<<Java Class>>
CommandsFactory
commands
CommandsFactory()
createCommand(String):ActionListener
<<Java Interface>>
ActionListener
java.awt.event
actionPerformed(ActionEvent):void



Class Name: MainApp.java	
Responsibilities: ▪ Initialize the application	**Collaborations:** ▪ GUI.java ▪ .... ▪ ....

Class Name: GUI.java	
Responsibilities: ▪ Handle drawing of user interface ▪ Handle user input	**Collaborations:** ▪ MainApp.java ▪ TextLineNumber.java ▪ Classes in the commands package

Class Name: TextLineNumber.java	
Responsibilities: ▪ Dynamically and automatically number the lines of the text editor so the user can choose specific lines with ease	**Collaborations:** ▪ GUI.java

Class Name: CommandsFactory.java	
Responsibilities: ▪ Create the required class objects to add functionality to the UI buttons and implement the ActionListener interface	**Collaborations:** ▪ GUI.java ▪ ActionListener Interface ▪ The rest of the classes in the commands package

Class Name: FileNew.java	
Responsibilities: ▪ Handle creating new internal documents	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: FileOpen.java	
Responsibilities: ▪ Handle opening files from the disk	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: FileSave.java	
Responsibilities: ▪ Handle saving files to the disk	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: FileEdit.java	
Responsibilities: ▪ Handle saving text changes from the editor into internal documents	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: TextPlayAll.java	
Responsibilities: ▪ Handle transforming the text from the editor into speech	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: TextPlayLine.java	
Responsibilities: ▪ Handle transforming a line of text from the editor into speech.	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: EncodedPlayAll.java	
Responsibilities: ▪ Handle encoding and transforming the text from the editor into speech	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: EncodedPlayLine.java	
Responsibilities: ▪ Handle encoding and transforming a line of text from the editor into speech	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: EncodeStrat.java	
Responsibilities: ▪ Handle choice of encoding strategy	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: ReversalToggle.java	
Responsibilities: ▪ Handle enabling and disabling the reversal of editor text	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: AudioVolume.java	
Responsibilities: ▪ Handle changing the audio volume according to user input in the UI	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: AudioPitch.java	
Responsibilities: ▪ Handle changing the audio voice pitch according to user input in the UI	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: AudioRate.java	
Responsibilities: ▪ Handle changing the audio speech rate according to user input in the UI	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes in the model package

Class Name: MacroManager.java	
Responsibilities: ▪ Handle the recording and re-execution of user defined commands	**Collaborations:** ▪ GUI.java ▪ CommandsFactory.java ▪ Classes that correspond to button functionality in the commands package

Class Name: DocumentInternal.java	
Responsibilities: ▪ Handle the storage and modification of the UI editor text	**Collaborations:** ▪ Line.java ▪ TextToSpeechAPIFactory.java ▪ EncodingStrategiesFactory.java ▪ Classes in the commands package

Class Name: Line.java	
Responsibilities: ▪ Handle the storage and modification of a single line of text	**Collaborations:** ▪ InternalDocument.java ▪ TextToSpeechAPIFactory.java ▪ EncodingStrategiesFactory.java ▪ Classes in the commands package

Class Name: TextToSpeechAPIFactory.java	
Responsibilities: ▪ Handle the creation of TextToSpeechAPI class objects required for the transformation of text to speech	**Collaborations:** ▪ DocumentInternal.java ▪ Line.java ▪ TextToSpeechAPI.java ▪ Classes that implement the TextToSpeechAPI Interface (currently FreeTTSAdapter.java) ▪ Classes in the commands package

Class Name: TextToSpeechAPI.java	
Responsibilities: ▪ Handle the interfacing of Text To Speech API adapters with the rest of the program.	**Collaborations:** ▪ TextToSpeechAPIFactory.java ▪ Classes that implement the TextToSpeechAPI Interface (currently FreeTTSAdapter.java) ▪ Classes in the commands package

Class Name: FreeTTSAdapter.java	
Responsibilities: ▪ Handle the adaptation of program data into a form that FreeTextToSpeech can use to provide text to speech transformation	**Collaborations:** ▪ DocumentInternal.java ▪ Line.java ▪ TextToSpeechAPI.java ▪ TextToSpeechAPIFactory.java ▪ Classes in the commands package

Class Name: EncodingStrategiesFactory.java	
Responsibilities: ▪ Handle the creation of class objects that handle text encoding	**Collaborations:** ▪ DocumentInternal.java ▪ Line.java ▪ EncodingStrategy.java Interface ▪ Classes that implement functionality defined in the Abstract class EncodingTemplate.java (currently EncodingAtbash.java and EncodingROT13.java) in the model package ▪ Classes in the commands package

Class Name: EncodingTemplate.java	
Responsibilities: ▪ Handle the implementation of a general text encoding method defined in the EncodingStrategy.java Interface and the definition of an abstract method that each encoding strategy will implement differently	**Collaborations:** ▪ EncodingStrategy.java Interface ▪ EncodingAtbash.java ▪ EncodingROT13.java ▪ Any other class that implements a specific encoding strategy method in the model package in the future ▪ Line.java ▪ DocumentInternal.java

Class Name: EncodingROT13.java	
Responsibilities: ▪ Handle the implementation of the abstract mapCharacter method defined in the EncodingTemplate.java Abstract class so that text can be encoded using the ROT13 algorithm	**Collaborations:** ▪ EncodingTemplate.java ▪ EncodingStrategiesFactory.java

Class Name: EncodingROT13.java	
Responsibilities: ▪ Handle the implementation of the abstract mapCharacter method defined in the EncodingTemplate.java Abstract class so that text can be encoded using the Atbash algorithm	**Collaborations:** ▪ EncodingTemplate.java ▪ EncodingStrategiesFactory.java