Text2SpeechEditor

Sprint No 1 Report

ΕΡΟΝΥΜΟυS ΙΩΑΝΝΗΣ ΧΟΥΛΙΑΡΑΣ , 2631 ΠΕΤΡΟΣ ΖΑΜΠΕΛΗΣ, 2970 ΔΗΜΗΤΡΙΟΣ ΖΩΓΑΝΑΣ, 2977

MYY803 PROJECT - 2020

VERSIONS HISTORY

Date	Version	Description	Author
1 may 2020	1.0	Last version of the requirements.	Team - Eponymous

1. Introduction

This document defines the basic requirements for Text2SpeechEditor, an application for writing documents and transforming them to audio.

1.1. Purpose

The objective of this project is to develop a simple editor that allows preparing text documents and transforming them to audio. Such an application can be of much help for people who have speaking problem. Moreover, it could be a useful tool for preparing presentations that are automatically played without a physical speaker. The application consists of a graphical user interface, a back-end that enables the text editing process and the part that transforms text to speech, which will be based on external text to speech libraries.

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

<For the user stories included in this release specify below corresponding tests using a typical tabular form.>

2.1. Scrum team

Product Owner	Apostolos Zarras
Scrum Master	Ioannis Chouliaras 2631
Development Team	Ioannis Chouliaras 2631, Petros Zampelis 2970, Dimitrios Zoganas 2977

2.2. Sprint Backlog

- [US1]: 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.
- [US2]: As a user, I want to edit the contents of the document, via the application's user interface

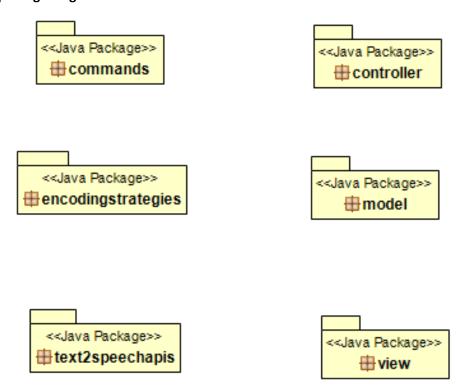
- [US3]: As a user, I want to save the contents of the document to disk by providing a particular filename. The application should automatically record the save date.
- [US4]: As a user, I want to open the contents of an existing document from disk by providing a particular file path, or by browsing the file system folders.
- [US5]: As a user, I want to transform the contents of the document to speech.
- [US6]: As a user, I want to select a line and transform it to speech.
- [US7]: 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.
- [US8]: 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.
- [US9]: As a user I want to encode the contents of the document and then transform them to speech.
- [US10]: As a user I want to select a line, encode it and transform it to speech.
- [US11]: As a user I want to tune the text encoding technique. In particular the application should support at least the following encoding strategies: AtBash and Rot13
- [US12]: As a user I want to be able to tune the audio parameters, i.e., the volume, the speech rate and the pitch.
- [US13]: 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.

•

3. Design

3.1. Architecture

UML package diagram:



Commands Package: contains all the classes for the actions/commands

Controller package: contains the class where the GUI(view) and back-end(model) communicates.

Model package: contains the back end of the application.

EncodingStrategies package: contains the interface and the classes for the two different encoding strategies.

Text2SpeechAPIS package: contains the interface and the classes for the transformation of the text to speech (freeTTS library)

View package: contains two classes. The GUI of the application(collaborating with Controller class) and the fileTypeFilter.

UML class diagrams:



Class Name: FileTypeFilter Responsibilities:

- Accepts the correct file.
- Extension .tts
- Opening and saving file to disk

Collaborations:

- openDocument
- saveDocument

Class Name: Text2SpeechEditorView Responsibilities: **Collaborations:** Whole GUI Controller package Interaction with user <<Java Class>> Ocument model encodingStrategy: EncodingStrategy audioManager: TextToSpeechAPI ttsFactory: TextToSpeechAPIFactory a authorString: String a titleString: String a dateString: String a lastModifiedString: String CDocument(String,String) <<Java Class>> playContents():void **G**Line playReverseContents():void model playEncodedContents():void encodingStrategy: EncodingStrategy playLine(int):void audioManager: TextToSpeechAPI playLineString(String):void uttsFactory: TextToSpeechAPlFactory playReversedLine(int):void □ line: String playEncodedLine(int):void w ords: ArrayList<String> tuneEncodingStrategy(EncodingStrategy):void setListFromText(String):void 0...* playLine():void replaceContents(String):void playReverseLine():void giveModifiedDate():void playEncodedLine():void getTitle():String tuneEncodingStrategy(EncodingStrategy):void getAuthor():String getLine():String getDate():String setVolumeLine(int):void getMDate():String setPitchLine(int):void setAuthor(String):void setRateLine(int):void setTitle(String):void setVolumeDocument(int):void setRateDocument(int):void setPitchDocument(int):void getPitch():int getRate():int aetVolume():int getArrayList():ArrayList<Line> getText():String setModifiedDate(String):void

Class Name: Line Responsibilities: **Collaborations:** Keep the words of one line

infoToWrite():String infoButton():String

- Play the words, set pitch, volume, rate
- Encode or play in reverse

- **EncodingStrategy**
- **TextToSpeechAPI**
- **TextToSpeechAPIFactory**

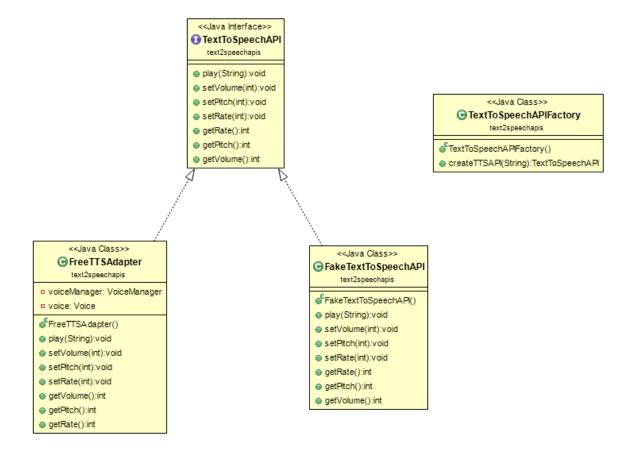
Class Name: Document

Responsibilities:

- Keeps Document info
- Contains a list of Lines
- Has all the methods to change the document fields
- Play the whole text in reverse etc

Collaborations:

- EncodingStrategy
- TextToSpeechAPI
- TextToSpeechAPIFactory
- Line



Class Name: TextToSpeechAPI

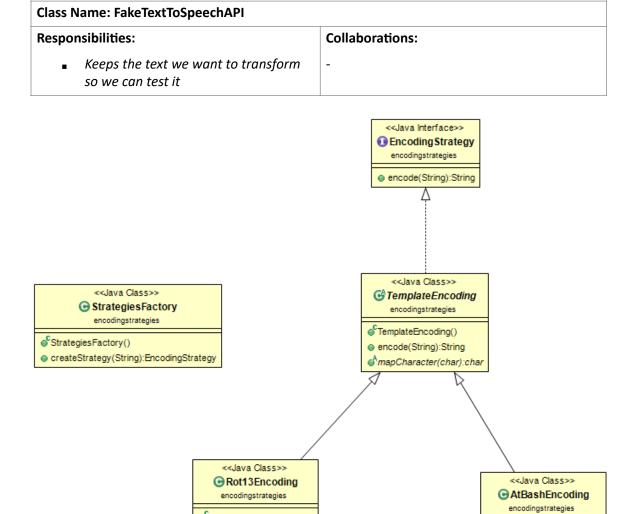
Responsibilities:

 Interface with some methods that FreeTTS and FakeTextToSpeechAPI can implement

Collaborations:

- FreeTTSAdapter
- FakeTextToSpeechAPI

Class Name: FreeTTSAdapter				
Responsibilities:	Collaborations:			
To transform the text to speech with the library FreeTTS	-			



Class Name: EncodingStrategy				
Responsibilities:	Collaborations:			
 Interface with one method. The abstract class TemplateEncoding will implement 	■ TemplateEncoding			

CRot13Encoding()

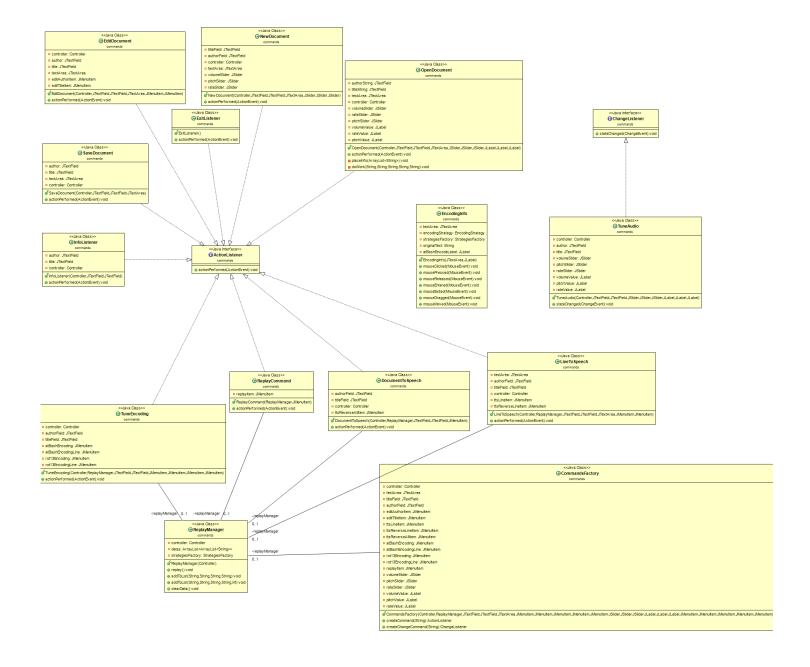
encode(String):String

mapCharacter(char):char

encode(String):String

mapCharacter(char):char

Class Name: TemplateEncoding				
Responsibilities:	Collaborations:			
 Abstract class where the encoding 	■ Rot13Encoding			
strategies will extend	AtBashEncoding			



Some commands are implementing ActionListener interface. TuneAudio implements a changeListener interface. Some commands collaborating with replayManager so we can store their actions.