German University in Cairo

Advanced Media Lab, 9

DMET 904 / CSEN 903

Project	Mini Project 1		
Group	Tutorial 9		
Team ID	Team 4		
Team Members	Member 1	Name	Amr El Zanaty
		ID	25-8370
	Member 2	Name	Hussien Mohamed Eloy
		ID	25-7532
	Member 3	Name	Ahmed Abd El-Naiem Elassuty
		ID	25-6759

In this mini project a game that recognises german words was implemented, It is mainly based on speech recognition tools, In this project the speech recognition was implemented using Sphinx4 Software tool where it is a software that recognises and manipulates speech through Java files , grammar file with extension .Gram , Dictionary file with the extension .Dict, a manifest file and configuration.xml file.
The following is the Dictionary that was implemented for the German Alphabet.

A	AH
B	B EH
C	T S EH
D	D EH
E	EY
F	EH F
G	G EY
H	HH AA
I	IY
J	Y OW T
K	K AA
L	EH L
M	EH M
N	EH N
O	OW
P	P EH
Q	K UW
R	EH R
S	EY S
T	T EY
U	UW

V	F AW
W	V EY
X	IH K S
Y	EH P S AH L AA N
Z	T S EY T
Ä	EY
Ü	EY UW
Ö	EY OW
ß	EY S EY T

Moreover we implemented a grammar file which constructs the grammar of the letters. It is basically the german letters that would be recognised by our project.

```
grammar letters;

public <letters> = <letter>;

<letter> = (A | B | C | D | E |F | G | H | I | J | K | L | M

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Ä | Ü | Ö |
ß);
```

Also the germanletters.java file which is the core of our mini project where it has the engine and the logic part for the recognition process, Also the code for the GUI.

This Javafile has the following methods:

- createGUI(): This method is responsible for the appearance of the User Interface of the game in which the user is asked to press a button to pronounce the letters and also where he is shown the time of playing and the letters he pronounce and the Original word he should pronounce and the recognised word by the Game, Moreover he is shown the score of the Game and the level he is playing at this moment.

    - HandleNextCharacter(): Where it handles and maps the characters recognised with the character at the corresponding place of the original word , where it checks for the correctness of this character and wether it is that same as the original word and if not it gives the user Three trials for each character through the variable “letter_trials” and then updates the score according to the recognition, Also it handles the levels of the game.

- GermanLetters(): This is the constructor of the where the GUI is created and the levels are added to the instance, And the Configuration Manages is initialised as well as the Recogniser class and Microphone class.

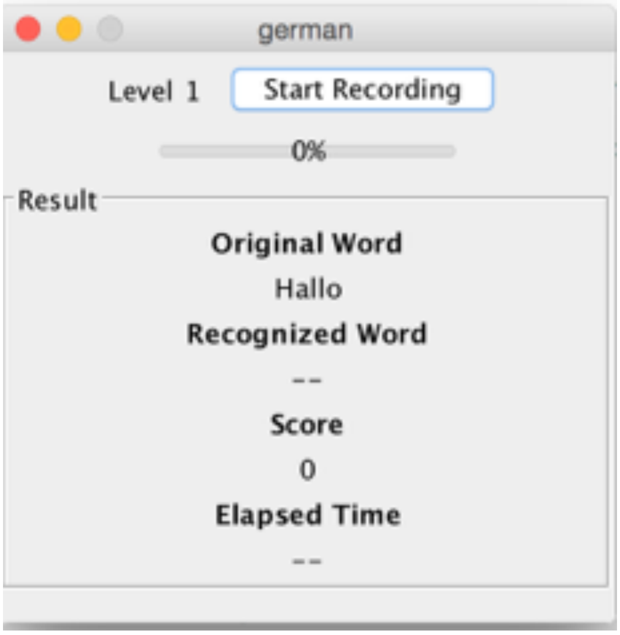
Moreover the GermanLetters Class has the Array that holds the Values of the original words that should be recognised at each level of the Game for example the words of the first level.

```
static String[] level_1 = { "Hallo", "Bluse", "Hund", "Aufwiedersehen",
"Tschuss", "danke", "bitte", "schÖn", "schlafen", "bett", "mund",
"hose", "nase", "schuhe", "singen", "sonne", "mond", "sterne","wolken",
"himmel" };
```

__Dealing with the application :__

First of all the player is viewed a original word and then the player is asked to pronounce the spelled letters of this specific word, Then the user should press start recording and hence he has three trials for each letter and the score of each level is calculated and updated regarding to the correctness or incorrectness of the specific character, Also a user is provided with the recognised characters by the application so that he is could know if he is pronouncing the right character or not, Moreover the player is viewed a progress bar that corresponds to where the player is from completing a level, and elapsed time which shows the time to the player .

And the following is a screenshot from the initial state of the game when the user is first presented the Game



Level 1

0%

Result

Original Word
Hallo
Recognized Word
--
Score
0
Elapsed Time
--