Final Report

*English to German Translator.*

*Project by Craig Methven, Elliot Kinkead, AJ Smart and Fraser Rossetter*

**Contents**

Introduction ………………………………..Page 2

Functional Requirements………………... Page 2

Non-Functional Requirements…………...Page 3

Use Cases………………………………….Page 3-4

Pseudocode………………………………..Page 5

Class Diagrams……………………………Page 6-9

Design of GUI……………………………...Page 10-11

Test Plan……………………………………Page 12-13

Task Allocations…………………………..Page 13

User Manual……………………………….Page 13-14

Evaluation………………………………….Page 15

**Introduction**

We have been asked to create a program which translates a text to/from English into/from another language of our choice (German). It should be able to read a file in

English or German, translate the file to the other language and print the translation to the screen and to another file. The program should also be able to take in a user input of a word or phrase and translate it to the screen and write it to a file. The user should be able to add and delete words or phrases to/from the dictionary file. When the program runs, it reads the dictionary file and creates a binary tree to store the words from the file alphabetically, so that when translating large files the Tree would be traversed efficiently. When the program translates a word or phrase, it either reads a .txt file or takes in user input, searches the binary Tree for the TreeNode with the corresponding String or Strings, outputs the translation in the other language and prints it to the screen and to a file of the users choice. When deleting a word or phrase to the dictionary, it traverses the binary tree using inorder traversal to search alphabetically for the TreeNode with the corresponding String or Strings and deletes it, both from the Tree and from the dictionary file.

Sources for German words: <http://frequencylists.blogspot.com/2015/12/the-2000-most-frequent-german-nouns.html>

[http://en.wiktionary.org/wiki/Wiktionary:Frequency\_lists/top\_2000\_German\_Wikipedia\_words](https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/top_2000_German_Wikipedia_words)

Sources for German phrases:

<https://pastebin.com/pqZ66NKJ>

Statement of requirements

Create a program allowing people to translate sentences from English to German, or German to English

Users: People learning German or English, likely over 10 years old.

Assumptions: The user is probably not a fluent speaker of one of the languages.

Inputs: The user inputs the sentence they want translated then chooses which way they want their sentence to be translated

Outputs: The translated sentence, buttons to select the option of which language to translate to/from

**Functional Requirements**

R1 Shall read in a file equivalent to a translator dictionary

R2 Shall take in user input

R3 Shall output the translated sentence

R4 Shall have buttons to select which language to translate from/to

R5 Shall translate both words and phrases into opposite language

R6 Should be able to add to dictionary if the word doesn’t exist

R7 Should be able to delete something from the dictionary

R8 Should be able to display the dictionary

R9 Should write to a new file for the dictionary if it has been edited

R10 May have a GUI

**Non-Functional Requirements**

NR1 Should be able to run outside of the IDE

**Use cases**

|  |  |  |
| --- | --- | --- |
| **Java Translator | Use Cases** | | |
| 1 | System: | Loads Java Translator. |
| 2 | System: | Import dictionary or ask user to load the dictionary. |
| 3 | User: | Loads Dictionary or uses default dictionary. |
| 4 | System: | Loads Binary Dictionary as well as phrase Linked lists |
| **Translate user inputted sentences.** | | |
| 1 | User: | Translate basic words |
| 2 | System: | Ask user if translating from to English to German or German to English? |
| 3 | User: | Chooses English to German |
| 4 | System: | Ask user to input |
| 5 | User: | User Inputs Translation. Exempel “Guten Morgen mein Name ist Jürgen und ich bin zwanzig Jahre alt” |
| 6 | System: | Then system prints to console the correct translation. |
| **Translate a text document** | | |
| 1 | User: | Selects translate text document |
| 2 | System: | Ask user if translating from to English to German or German to English? |
| 3 | User: | Selects English to German or vice versa |
| 4 | System: | Will either save the translated text as or outputs it to the screen |
| **Add words to the dictionary** | | |
| 1 | User: | Selects add words to the dictionary. |
| 2 | System: | Offer a choice between adding a word list by selecting a word list or allow the user to input the words manually |
| 3 | User: | Choices either option and inputs the words he would like to input to the dictionary along side it’s German translation. |
| 4 | System: | Add words to the dictionary |
| **Removes word from the dictionary.** | | |
| 1 | User: | User selects to remove words from the dictionary |
| 2 | System: | Ask user what word they would like to remove |
| 3 | User: | Types the word they would like to remove |
| 4 | System: | Searches for the word that the user inputted. If it can’t find that word it will notify the user, they can’t find that word.  If it finds the word, then it’s asked user. Are they sure they would like to delete the word? |
| 5 | User: | Selects Yes |
| 6 | System: | Deletes the word from the dictionary. |

**Psuedocode**

Finding German word from English one:

Method that returns String and takes in **int** findEngWord and TreeNode currentNode{

//If it is the item we are looking for then return current node

**if**(currentNode English word == findEngWord {

**return**(The german word that is saved in the current Node);

}

//if it is greater than the current node look right

**else** **if**(currentNode’s English word is later in the alphabet than findEngWord) {

**if**(There is a node to the right) {

String tempString = findItem(findEngWord, the reference to the right node saved in current node);

**return**(tempString);

}

}

//if it is less than the current node look left

**else** **if**(currentNode’s English word is earlier in the alphabet than findEngWord) {

**if**(There is a node to the left) {

String tempString = findItem(findEngWord, the reference to the left node saved in current node);

**return**(tempString);

}

}

//If it reaches here then it doesn’t exist in the binary tree

Run method that allows the user to add a translation for the word that we don’t translate

**return**(“”);

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Class Diagram**   |  |  | | --- | --- | | **Menu** | | | translation | The instance of Translator that the menu is using | | englishDoc | The name of the file storing the english words | | germanDoc | The name of the file storing the german words | | germanPhraseDoc | The name of the file storing the english phrases | | englishPhraseDoc | The name of the file storing the german phrases | | engOrGer | Boolean storing if the menu should be in german or english | | main() | Starts the program | | Welcome() | Allows user to select menus language | | runMenu() | The main loop that loops all the options that the user has to choose from. The central navigation point | | findENGorGer() | Asks if the user is going from english to german or vise versa and sets this option in the translator class | | dictionary() | Another menu for the user to be able to select what they’d like to do with the dictionary | | translateText() | Takes in the name of a file and sends it to be translated | | addToDictionary() | Allows the user to add words to the dictionary | | deleteFromDictionary() | Allows users to remove words from the dictionary |      |  |  | | --- | --- | | **Translator** | | | translatedSentence[] | Array to store all the sentences that have been translated | | sentence[] | Array to store all the sentences that haven’t been translated | | gerwordTree | The tree reference for the german words | | engwordTree | The tree reference for the English words | | gerphraseTree | The tree reference for the german phrases | | engphraseTree | The tree reference for the English phrases | | skip | Boolean to tell if the program should ask the user for a translation if a word isn’t in the dictionary | | engOrGer | Determines if the translation if going from english to german or vise versa | | takeInFile() | take in the file containing the sentence to be translated | | output() | Sends the translation to a file | | outputToScreen() | Prints the translation to the console | | checkIfPhrase() | checks to see if what is passed is in the phrase list | | translateSentence() | Translated a sentence given a sentence | | takeInString() | Takes in a string to translate | | SeparateString() | separates the original sentence into an array which holds each word (and non word character), translates the word and saves it to the translated sentence array | | searchPhrases() | searches the phrase tree and returns the translation if it has one | | sendToTree() | finds the word from the sentence in one language tree and uses its index to find the translation in the other language tree | | findWordTree() | returns the correct tree corresponding to if the user is translating from english to german or german to english | | findPhraseTree() | returns the correct tree corresponding to if the user is translating from english to german or german to english | | addWord() | Adds a word to the dictionary | | printDictionary() | Prints the dictionary to the console | | deleteFromDictionary() | Deletes a word from the dictionary | | addWordToFile() | Adds a word to a file given |      |  |  | | --- | --- | | **Tree** | | | normalWordDoc | File name of the file that stores normal words of each of this trees nodes | | translationWordDoc | File name of the file that stored translated words of each of this trees nodes | | root | The root of the tree | | balanceFileOriginal | Name of a file used to balance the tree | | balanceFileTranslation | Name of a file used to balance the tree | | addNode() | Adds a node to the tree given 2 words | | printTreeInOrder() | This method prints the the tree in Inorder traversal | | printTreePreOrder() | This method prints the the tree in preorder traversal | | deleteNode(String) | This method deletes a node from the tree | | deleteNode(TreeNode) | This method deletes a TreeNode | | findItemStart(String) | Method to initialise finding a translation | | findTranslation(String, TreeNode) | This method returns the translation | | saveTreeForOneLanguage(int) | This saves the tree to a file | | readTree(int) | This methods reads in a file and saves it to the tree | | balanceTree() | Balances Tree |      |  |  | | --- | --- | | **TreeNode** | | | left | Node reference to the node to the left of this one | | right | Node reference to the node to the right of this one | | up | Node reference to the node above this one | | original | Stores the original word | | translation | Stores the translation of the word | | counterForSaving | Static variable used when saving the tree to an array | | printInOrder() | Print’s Tree inorder Traversal | | printPreorder() | Print’s Tree in PreOrder Traversal | | printNode() | Prints the node for the console | | countTreeSize() | This method counts the size of the tree using recursion | | saveTreeToArrayPreorder() | Saves the tree to an array in a preorder | | saveTreeToArrayPreorder() | Saves the tree to an array in a Inorder | | addNode(TreeNode, TreeNode) | This adds nodes to the tree. |   **Design of GUI**  Starting Screen    Menu |  |
|  | **Test Plans and Result**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test** | **Input** | **Expected Output** | **Actual Output** | **Additional Comments** | | Can it translate a document from English to German using words | Craig is awesome | Craig ist genial | Craig ist genial |  | | Can it translate a document from German to English using words | Craig ist genial | Craig is awesome | Craig is awesome |  | | Can it translate a document from English to German using phrases | Can she do it | Kann sie das | Kann sie das |  | | Can it translate a document from German to English using phrases | Kann sie das | Can she do it | Can she do it |  | | Can it translate a sentence via the console from English to German | Potato is tasty | Die Kartoffel ist lecker | Die Kartoffel ist lecker |  | | Can it translate a sentence via the console from German to English | Die Kartoffeln sind Lecker | the potatoes are tasty | the potatoes are tasty |  | | Can the translator program handle additional characters such as ä ö, ü and ß | Over size possible | über große möglich | über große möglich |  | | Can you add to the Dictionary | Elliot as Elliot | Elliot | Elliot | We weren’t prompted with the word wasn’t in the dictionary so it was added correctly | | Can you delete from the Dictionary | Elliot | A prompt to ask us for a translation | A prompt to ask us for a translation |  | | |
|  |  |  |
|  | **Task Allocations**  The allocations were the following.  Going between binary tree and menu - Fraser  Find file of foreign words - Fraser  Create binary trees for words - Craig  Input to things the binary tree can use- AJ  Menus and GUI – Elliot && Fraser  How it worked was the following.  Going between binary tree and menu - AJ, and a bit of Craig  Find file of foreign words - Craig  Create binary trees for words - Craig  Input to things the binary tree can use- AJ, and a bit of Craig  Menus and GUI – Elliot && Fraser, with corrections from AJ and Craig  **User Manual**  To use the program, you will first be presented with the option to use the program in English or in German. Input the language that you want to read the program in and the program will run in that language. To:  Translate a .txt file, **enter 1.**  Do something with the dictionary, **enter 2.**  Input something to be translated, **enter 3.**  End program, enter 4.  **1. Translate a .txt file.**  You will be prompted to choose whether to translate a .txt file in English to German or a German .txt to English. Then input your file you want to translate and the program will ask you to create a file name for your translated .txt file. Your original file will then be translated, saved into the new file and outputted to your screen.  **2. Do something with the dictionary.**  You will be presented with a menu to select from the following options:  To print the dictionary, input 1.  To add a word or phrase to the dictionary, input 2.  To delete from the dictionary, input 3.  To return to the main menu, input 4.  2. Add to the dictionary.  You will be asked if you want to translate a phrase or a word. To add a phrase, input 1 and to add a word, input 2. Then you will be prompted to input the English translation of the word/phrase you want to add, and then the German translation it. If your word/phrase is not already in the dictionary, it will be added and the program will tell you it was added successfully.  3. Delete from the dictionary.  You will be asked to input the word/phrase you want to delete from the dictionary, you can either input its English or German translation. If the word/phrase exists in the dictionary, it will be deleted and the program will tell you it was deleted.  .  **3. Input something to be translated.**  You will be prompted to input what you want translated. Enter the word/phrase you want translated and if it is in the dictionary, the program will translate it, print it to the screen and ask you to create a .txt file to store the translation. |  |

**Evaluation**

The final task has gone well and we were able to complete the project on time and covered all the minimum requirements as well as several extra features such offering a German version of a translator, translating phrases. We additionally wanted to do include a GUI however due to several communication issues we did run out of time.

We did encounter several issues whilst working on the project such as splitting words and punctuation. Splitting words were fine and worked perfectly, however splitting words that had punctuation such as “Wissen Sie, wo wir sind?” led to the program removing the punctuation. This took some time to solve however the current version of the program still include punctuation. Another issue that we encountered was checking the sentence and phrases mainly the original method that we used to check the sentence didn’t work and we had to devise a new method which involved spitting up the sentences to compare to the list of phrases which now means we have a slower translation time, however it does mean this now works.

Another issue we encountered was dealing with german umlaut characters such as ä, ö, ü and ß . We realised that we should be handling the text document using UTF-8 encoding rather than a standard text encoding. It was also quite difficult to balance the tree and the method we used doesn’t work well, therefore, we not running a balanced tree method. Other problems also included “?” in random places as well as getting a dictionary was a pain , mostly because there wasn’t a German dictionary readily available the internet, therefore, we had to make our own German dictionary and our own phrasebook. There were was one issue that was unable to resolve the first one was GUI was able to run but was throwing errors additionally because we only got the program running correctly closer to the deadline than we would have liked. We wouldn't have had enough time to get working and implemented, therefore, we haven't included it.

Overall, We completed all the minimum requirements and a few extra we also included a German version of the menu. However, due to communication and program issues as well as a lack of time we weren't able to create additional features.