B. Comp. Dissertation

**Talk-to-Code: Coding by Dictation**

By

Lam Zhen Zong Nicholas

Department of Computer Science

School of Computing

National University of Singapore

2017/2018

B. Comp. Dissertation

**Talk-to-Code: Coding by Dictation**

By

Lam Zhen Zong Nicholas

Department of Computer Science

School of Computing

National University of Singapore

2017/2018

Project No: H0201280

Project Supervisor: Assoc Prof. Ooi Wei Tsang

Deliverables:

Report: 1 Volume

Table of Contents

[1 Introduction 1](#_Toc497432426)

[1.1 Project Objectives 1](#_Toc497432427)

[2 Literature Review 1](#_Toc497432428)

[3 Progress made so far 1](#_Toc497432429)

[3.1 Project Overview 1](#_Toc497432430)

[3.2 Speech Recognizer Module 3](#_Toc497432431)

[3.2.1 Experiment with Basket of keywords 3](#_Toc497432432)

[5 References 3](#_Toc497432433)

# 1 Introduction

## 1.1 Project Objectives

Coding by Dictation is a continuation of the previous project ‘Talk-to-Code: From Structured Command to Source Code’ (Gao, 2016) which creates a foundation for creating a hands-free natural language programming application. This application allows users to write program code by just dictating the program to the computer via voice input.

This project aims to enable people with disabilities or conditions such as the Carpal Tunnel Syndrome (CTS) to write computer programs. CTS (also known as Repetitive Strain Injury) is a medical condition which affects the median nerves of the hand, causing pain and discomfort to the user. It is found that majority of people diagnosed with CTS tend to perform repetitive tasks of some sort involving the use of fingers or hands. (Crouch, 1995) Thus, it is of no wonder that programmers fall under this category, as they perform repetitive typing tasks to write computer programs.

This project will help programmers to write program code without using their hands, therefore reducing the risk of contracting CTS, or even enable CTS patients to write code once again.

# 2 Literature Review

# 3 Progress made so far

## 3.1 Project Overview

This project focuses on creating an end-to-end program which converts a user’s voice into C program code. The ideal situation would be to have a program which allows programmers to speak in a human and natural way, and translates that to code subsequently. However, it is noted that there can be many different ways to verbalize the same piece of code, and that natural English commands were rather ambiguous in the sense that there can be many different interpretations of that natural English command. (Chew, 2016)

For instance, in order to declare an integer variable with value zero, one programmer might verbalize “Create integer variable with value zero” while another programmer might say it as “Declare an integer variable equal zero”. Both programmers are trying to write the exact same piece of code through different ways of saying it, but it is not easy to have a program that is able to decipher and know that these two different verbalizations should result in the generation of the exact same piece of code.

As for the latter case regarding the ambiguity of the natural English commands, the English command “declare integer x, y, z” can be interpreted (in code) as “int xyz;” or “int x, y, z;”. This ambiguity can result in the 2 different interpretations as outlined above, and this causes problems as the program will not know which piece of code to generate and what the user actually intends to write.

In order to eliminate ambiguities, this project will introduce a structured and fixed way of expressing programming constructs, while incorporating some natural English in my structured language as well. The structured language will be presented in a later section in this paper. The system architecture of the Coding by Dictation application at the current stage is depicted in the diagram below.

Voice input from user

Structured Command

Processed text

English text (structured language)

Program Code

Structured Command Parser Module

Word Parser Module

Word Corrector Module

Speech Recognizer Module

## 3.2 Speech Recognizer Module

My Speech Recognizer Module converts voice input from the user into English text, using the python speech recognition library (Zhang, 2017). The python speech recognition library supports several APIs in which we can use to perform the speech recognition. Out of the several APIs, I have picked 3 of them (mainly Google Speech Recognition, Google Cloud Speech API, and Microsoft Bing Voice Recognition) to carry out further testing and experiment before deciding on one of them to use for my Coding by Dictation application.

### 3.2.1 Experiment with Basket of keywords

(perhaps covering design/investigation alternatives, and

thoughts on preliminary design/investigation)

Final Year Project (FYP)

Submission Guidelines for CA Report

The CA Report is a short report (10-15 pages) that covers:

 project title;

 project objectives description;

 literature review

 progress made so far (perhaps covering design/investigation alternatives, and

thoughts on preliminary design/investigation); and

 research plan for the next semester.

 list of references.

# 5 References

Chew, Y.X. (2016). Hands Free Programming. National University of Singapore, 2016

Crouch, Tammy (1995). Carpal Tunnel Syndrome and Repetitive Stress Injuries: The Comprehensive Guide to Prevention, Treatment and Recovery. Frog, Ltd. Berkeley, California.

Gao, R.S. (2016). Talk-to-Code: From Structured Command to Source Code. National University of Singapore, 2016

Zhang, A. (2017). Speech Recognition (Version 3.7) [Software]. Available from <https://github.com/Uberi/speech_recognition#readme>

 supervisors / main evaluators can have informal discussion about the student’s understanding of the project, progress, what he would be doing in the next semester, and expected outcome at the end of the project.

1. Read http://advice.writing.utoronto.ca/types-of-writing/literature-review/ to find out

what literature reviews should and should not be. You may want to check with your

supervisor on the best way to do a literature review in your research area.

2. Make sure you are aware of what constitutes plagiarism. See

http://www.cdtl.nus.edu.sg/success/sl7.htm.

3. You should submit one hard copy to your supervisor by the deadline

(Wednesday, week 13).

FYP report - interim can follow final report format  
Submit to supervisor as soft copy  
Check with evaluator submission

This is a gentle reminder for you that the deadline to submit your CA reports directly to your respective supervisor and main evaluator is **5.00pm on 8th November 2017.**