Oregon State University
College of Engineering

CS Capstone Winter Progress Report

February 10, 2018

Kora

Prepared for

Autodesk

Anand Karyekar

Jennifer Talaga

Prepared by

Group8

Austin Row

Jeremy Fischer

Abstract

This document outlines the production progress of Kora from December 2017 through the middle of February 2018, including obstacles and steps moving forward.

CONTENTS

1 Purpose

Kora is a proof of concept project that will integrate a natural language processing library into Autodesk's 3-D computer aided design software, Fusion. Kora will be a speech-based virtual assistant for Fusion that lets users perform a subset of tasks within the product, such as saving a document or opening a menu, by verbally instructing it to perform the task. As a stretch goal, Kora will be capable of questioning the user and using responses to predict and automatically assist with future user behavior.

Kora will offer users a tool that decreases the time required to achieve their goals within Fusion by offering an interface that runs in parallel with and complements the keyboard and mouse. If the stretch goal is achieved, Kora will further increase productivity by learning to automate specific workflows within the product.

2 Goals

2.1 Main Goals

- Kora will allow the user to perform tasks in Fusion via a voice interface. Each voice command given by the user will be mapped to a specific task which will then be performed in the open Fusion design.
- All interactions with Kora will be logged for debugging and future development purposes.

2.2 Stretch Goals

- Kora will periodically ask the user questions regarding why they have performed specific tasks. The questions asked by Kora will pertain to specific predefined contexts.
- Kora will record user responses and save a transcript along with contextual information.
- User responses to interview questions will be used to try to predict and assist with future user actions.

3 Project Progress

Kora's MongoDB database will have two collections: Interaction and Logs. The Interaction collection is setup and functioning. The Interaction collection stores documents describing each user-Kora interaction. Each document contains the posting date, the user's Fusion ID number, the JSON returned by WIT.ai, the chosen Fusion API call, Fusion's execution status, and the time it took from when the user stopped talking to the time Fusion finished executing the command.

Kora can now execute the two commands we marked as our baseline commands in the requirements document: rotate and save. A user can activate Kora and say any permutation of "Rotate $< direction >< number >$ degrees", "Rotate $< number >$ degrees $< direction >$", "Save this", or "Save as $< name >$." When the user is finished speaking Kora successfully executes the command within Fusion.

4 Obstacles

As per our requirements, Kora must function on both Windows and MacOS. This has caused subtle issues that were challenging to track down. An issue in particular was PyAudio; the library used to stream audio to Wit.ai. Kora and PyAudio worked fine on Windows. However, when trying to run Kora on Mac, Fusion would crash without error messages or hints, just crash. After some thorough investigating we found that the problem occurred because the default microphone sample rate on Windows is significantly lower than the default sample rate on Mac. This caused an overflow in PyAudio's streaming buffer resulting in Fusion crashing.

Kora being a voice assistant means it must be able to execute commands quickly. Users don't want to command Kora to do something and have to wait approximately seven and a half seconds before she accomplishes it. Unfortunately, that is Kora's current average speed. The problem is Wit.ai, our natural language processor, is on the web and Kora has to make HTTP calls to it. This on average takes seven and a half seconds. We are looking for a local alternative, but Wit.ai has all of Kora's learnings on their server which can only be accessed via HTTP.

Fusion Add-Ins and Scripts are run in their own environment. They have their own version of Python and their own Python modules. This has been tricky for two reasons. The first is pathing. Since Fusion's Python only looks for libraries within its own environment we either have to append the directory of our newly downloaded library to the system path, causing a bloated system path variable, or change all of the absolute import paths in the library to relative imports. To abide by good practices we have been changing the import paths to relative paths. The second is library versions. We have ran into issues where we pip install a library and it downloads the Python 2.7 version since that is what we are using locally, and when we move the library into Fusion, Fusion crashes because Fusion's built in Python is Python 3.3.

5 Moving Forward

As mentioned above in the Obstacles section, we have a big problem with Kora's latency due to HTTP calls to Wit.ai's servers. Our top priority moving forward is to bring the natural language processing library to our local machine. A significant amount of our code is built upon the response returned from Wit.ai, so the sooner we have a solution to our latency problem the smaller headache we will have further down the road when refactoring our code to work with the solution. We aren't aware if this is possible with Wit.ai. We may need to pivot and use a new library altogether.

Moving forward we will also be integrating a suite of unit tests to verify Kora's robustness. These tests will help us locate any weak points or subtle bugs which reside in Kora, as well as ensure that any new code submitted doesn't introduce new issues.

6 Photos

The assignment page said: *includes images of your project – screen shots, photos, whatever is appropriate.* So, maybe we could add a picture of the tool Add-In drop down that says "Activate/Deactive Kora", and maybe a Fusion message box popping up with something relevant on it.