CS Senior Capstone Fall Term - Project Report

December 2nd, 2017

CDK Global: No more touch. No more Keyboard. Bring it All Together. Using Technology to Teach Humans.

Prepared for

CDK Global

Trevor Moore ______________________ Signature ________ Date

Prepared by

Group 9

Look Boss, No Hands

Brandon Dring ______________________ Signature ________ Date

Nipun Bathini ______________________ Signature ________ Date

Carl Benson ______________________ Signature ________ Date

Abstract

This document seeks to explain what exactly we have been doing for the past 10 weeks with regards to the Look Boss, No Hands project. We will go over exactly what the project is trying to accomplish, and the weekly status updates on progress that lead up to where we are now. Then, a short excerpt talking about where the project is in its current form. Afterwards we will explain any problems we have encountered so far, and any interesting findings so far. Lastly, we will go over a retrospective on what we've done as a team this semester, and ways we can improve.

CONTENTS

1 Project Purpose and Goals

Look Boss, No hands seeks to develop a new interaction method between a user looking at enterprise data. Instead of sitting behind a computer screen typing and clicking around to get graphs, and charts. This project aims to remove the standard methods of input, in lieu of more novel methods of input such as, voice and virtual reality. Using new pieces of hardware taking over the technology industry, a user should be able to tell a Alexa what they want to see, and have it displayed for them in VR to interact with

2 Weekly Progress

2.1 Week 2: Housekeeping, introductions, client meeting planning

2.1.1 *Overall*

Week 2 was the week that we received our project assignments. We set up a Slack channel to maintain communication and invited our client to it as well. Brandon had created the Github for all the project related stuff, along with Google drive for sharing. We were in contact with our client and scheduled a time to have a face-to-face meeting the next week.

2.2 Week 3: Client meeting, Team Meeting, Problem Statement

2.2.1 *Client*

Week 3 featured a meeting with our client and the rough draft of the problem statement. We met with our client for lunch and went over more details of the project as well as discussed what technologies we would be using, along with the hardware required. We also set up a recurring, weekly meeting over WebEx to keep our client up to date with current progress.

2.2.2 *Writing*

We finished and submitted the rough draft for the problem statement. In the problem statement, our description of the problem was that there are many new types of devices being introduced that lacked a connection between them.

2.3 Week 4: AWS setup, create Alexa, Lambda, Problem statement

2.3.1 *Writing*

Week 4 included the submission of our final draft for the problem statement as well as the beginning of experimentation with custom Alexa commands. We took the feedback from the rough draft as well as additional proofreading to fine tune the final draft for submission

2.3.2 *Project*

Brandon had set up the first instances of accounts allowing for the creation of Alexa skills and processing.

2.4 Week 5: Requirements Document, Alexa and AWS

2.4.1 *Overall*

Week 5 was the week we begin the requirements document in addition to beginning investigation into created custom Alexa skills. We each created a simple new command for Alexa to acknowledge and respond to.

2.4.2 Writing

We wrote the rough draft for the requirements document. In the requirements document we went over more in depth details, specific requirements, and other non-functional requirements. The rough draft was sent to our client for feedback and approval.

2.4.3 Project

Alexa and AWS lambda were also programmed to connect to the remote server and get some data and return it.

2.5 Week 6: Requirements document, TravisCI, Refactor, Socket.IO

2.5.1 Writing

Week 6 was similar to the previous week. We revised our requirements document based on provided feedback and received client approval. We continued our work with custom Alexa commands, increasing functionality through more recognized phrases.

2.5.2 Project

Created ability to connect websockets on the remote server and have them broadcast messages to local laptop. Created continuous integration testing on Github for each new commit and PR. Refactored AWS files and folders to be more modular and readable.

2.6 Week 7: Tech Review, VR headsets

2.6.1 Overall

Week 7 primarily focused on each of us beginning our technology review document. Brandon reviewed options for server coding language, graphics engine, and network communication framework. Nipun reviewed options for virtual assistants, data storage, and wearables. Carl reviewed options for VR headsets, data hosting services, and machine learning frameworks.

2.6.2 Project

Client had received VR headsets, implemented basic game in Unity in VR to play with blocks and throw them around.

2.7 Week 8: Agile adoption, MongoDB mock data

2.7.1 Writing

Week 8 featured the editing and submission of the final drafts of our technology reviews. We took the feedback we each received from the in-class peer review to identify the areas we needed to make changes.

2.7.2 Project

Everyone created agile stories for tasks relating the project. Defined sprint lengths, and set up more regular meeting times and demos for the project with client. Created script to create mock sales data to be used in VR later on in the project.

2.8 Week 9: Unique Alexa commands

2.8.1 Project

Week 9 was the week we began our Agile development. Brandon setup Zenhub to add, track, and assign stories. The stories for this sprint were:

- Nipun created a comprehensive README for both the Alexa and AWS server setup as well as created Mocha tests for basic server routes.
- Carl researched machine learning frameworks for providing more human-like responses from Alexa as well as created a Python script for the generation of fake sales data.
- Brandon created a basic VR environment in Unity as well as looked into the implementation of RabbitMQ in Unity.

Created unique Alexa commands by registering variables in the sample utterances that later tell the server what route to get and what data to query.

2.9 Week 10: Sprints, Progress report, design doc

2.9.1 Writing

Week 10 included the writing of the design document and the progress report. The design document detailed the design and implementation details, as well as information on the components of the system from multiple viewpoints.

The other task we worked on this week was the progress report, which covers what we accomplished this past term in the form of a written report and a recorded presentation.

3 Where We Are

Currently we have set up the skeleton framework of our project, meaning that we have have created a functional Alexa account that can take in basic commands and phrases. We have also a pretty solid understanding as how to modify and add onto what we currently have with the Alexa to add more commands and use speech variables within our sample utterances.

On top of that, we have a functional Lambda service inside AWS that the Alexa speech processing maps to. From there we have configured certain routes to use variables dictated by the speech patterns received by the Alexa and go to the respective server routes to query the correct data. Along with querying routes on the server, there is also semi-intelligent responses whether an error was encountered or Alexa/Lambda couldn't understand the request.

As for the VR side of things, it wasn't until about a month ago that we actually had a VR headset in hand to begin development. But since acquiring the headset, we have researched packages to use to fit our needs and implemented it into a working skeleton platform. We have found a very helpful package called VRTK that helps abstract away all the complexity of the math and physics that goes into creating a VR experience. As such, we currently have a VR environment where the user can move around and teleport, interact with objects and get logical feedback regarding physics, and even climb trees. We have also connected the local VR headset up to the AWS server, so that the commands received from the Alexa eventually make its way all the way down to the local headset.

4 Problems, and solutions

One issue we ran into was waiting for the hardware

- Waiting for the Alexa was not that bad, and we received them on October 18th.
- But waiting for the HTC Vive until the middle of November, made it difficult for us to begin to get a start on the VR side

We also did not all setup Zenhub, until the start of November, and assigning tasks has been much simpler since.

The client had also requested Alexa seem non-robotic, thus the responses shouldn't all be hardcoded and Alexa should provide useful suggestions to the user.

- This makes us think of using another AWS service for machine learning, or switching part of the project over to Python to use some of their famous AI/Machine Learning libraries.

Using something simple in VR like data visualization has already been done. But the package/asset requires money to buy.

- We're stuck at either developing our own data visualization library, or convincing the client to buy some.

The amount of writing that has been required by the class so far has taken away from the time to actually build the project

- Significant work will be implemented over the course of winter break to make up for lost time during fall term.

Instead of sharing an AWS account (which requires a credit card), to avoid others seeing private financial information. We have all had to create separate AWS accounts and clone the code from Github and paste it into whatever AWS service we need.

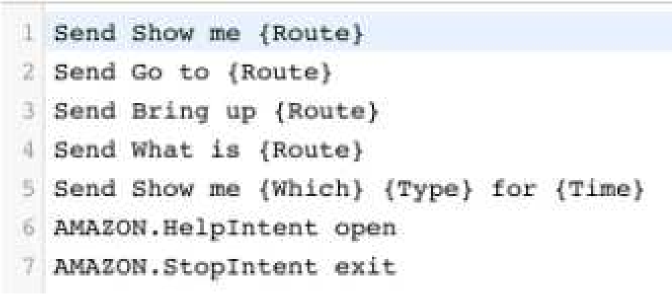
- As a solution, I might be able to open up an AWS account since it is free for 12 months with a prepaid credit card that I has $3 on it. Then everyone can share the same account and not worry about copy and pasting code.

5 Interesting Code

The way that the Alexa SDK is set up inside AWS is really quite ingenious, allowing for rapid development and quick modifications.

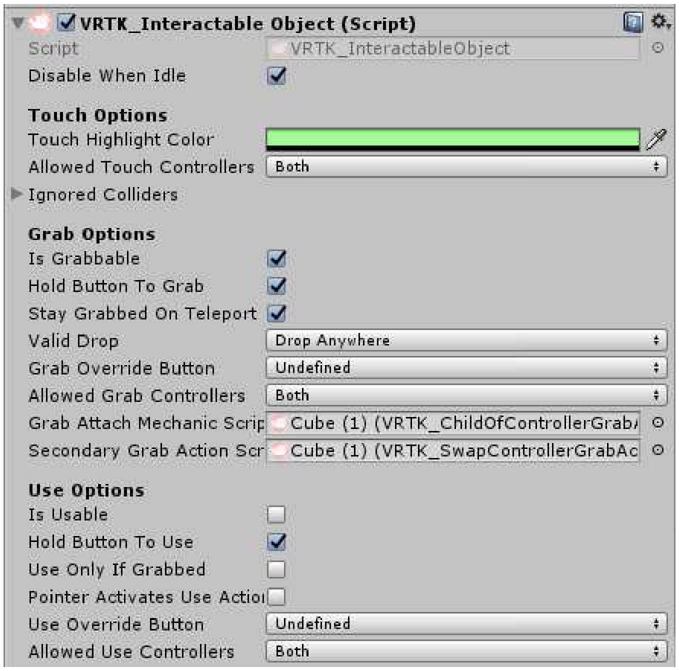
For example, figure 1 shows how you specify to Alexa what commands to listen for within your custom Alexa skill. The first word is the intent type/name, the following words after that are the exact commands that Alexa listens for. The values wrapped in {} are variables which you can assign any value, which are later passed into an object with the key Route along with whatever value filled in the variable Route.

The Unity package that we had found was also really neat as well, VRTK allows for simple drag-and-drop attributes onto objects within the VR environment. Where you search for the functionality that you desire, and drag it onto the GameObject you want. There is no longer a need for computing angular velocity when throwing an object or dropping it.



```
Send Show me {Route}
Send Go to {Route}
Send Bring up {Route}
Send What is {Route}
Send Show me {Which} {Type} for {Time}
AMAZON.HelpIntent open
AMAZON.StopIntent exit
```

Fig. 1: Sample Alexa utterance list



☑ **VRTK_Interactable Object (Script)**

Script VRTK_InteractableObject

Disable When Idle ☑

Touch Options

Touch Highlight Color 🟩

Allowed Touch Controllers Both

Ignored Colliders

Grab Options

Is Grabbable ☑

Hold Button To Grab ☑

Stay Grabbed On Teleport ☑

Valid Drop Drop Anywhere

Grab Override Button Undefined

Allowed Grab Controllers Both

Grab Attach Mechanic Scrip Cube (1) (VRTK_ChildOfControllerGrab/

Secondary Grab Action Scr Cube (1) (VRTK_SwapControllerGrabAc

Use Options

Is Usable □

Hold Button To Use ☑

Use Only If Grabbed □

Pointer Activates Use Actio □

Use Override Button Undefined

Allowed Use Controllers Both

Fig. 2: Object attributes with VRTK

Figure 2 shows the attributes put onto a cube, and the added options of it being grabbable and usable. This package allows us to not reinvent the wheel, and use something that is probably both more reliable and efficient when it comes to VR interactions.

6 Retrospective

Positives	Negatives	Actions
All pieces of required hardware were received	Lack of planning and forethought with regards to Agile throughout start of semester	Creating future tasks, breaking them down into respectable size, and greater accountability between group members for project tasks
All necessary software frameworks were set up	Separate AWS accounts	Create a shared AWS using pre-paid credit card
Significant contribution from all regarding group writing assignments	Sharing single VR headset	Brandon bought a personal Oculus Rift during Black Friday
Base functionality of all aspects of the project were completed	Difficult to demo for client	Pre-record demos to be posted to Google Drive or ENGR webspace