MorphoGenesis

Team S.H.A.P.E

Aaron Kwan

Andrew Lewis

Jonah Simon

Nick Strazis

Patrick Bowling

Table of Contents

Executive Summary

As of October 27th 2017, the prototypes for the project are close to being the final product. The Emotive headset, Affectiva, Server, Android App, Kinect, and Alexa are in the final stages of development and need either only cleanup or one additional feature before they become the finished product. To ensure that the prototypes are functioning properly, the team brought in users to test each of the products. The results were that each prototype functioned properly, but some of the prototypes such as the kinect were not as user friendly as the team was expecting. As such, the team will spend some time fixing this problem before the product becomes finalized.

Results to Date

Beta Prototype Description Update

Affectiva

As of October 27th, the Affectiva face recognition software is able to take in a user's face and determine which emotion is being presented. The software is able to be used on any webcam and can take that data and transmit it to the server. The Affectiva still needs to support a few more emotions, but otherwise the prototype is close to being the final product

Alexa

Our voice recognition prototype can successfully receive commands through Alexa when the user gives either a configuration ("Configuration 2") or an emotion ("I am angry", "I feel scared"). It still has the problem of crashing when it receives an invalid command instead of prompting the user for a valid command. We are currently in the process of fixing this. Another problem we have encountered with Alexa is that it will exit the SHAPE Wall skill after not receiving any input for ~10 seconds. Unfortunately, we do not have a solution for this; Alexa has a command built in to close out of a skill after a set period of time without input, and this time cannot be altered. The only possible fix would be to set a delay in the skill itself after receiving each command, but this would prevent the user from giving input during this delay.

Android App

The Android application (see image 1) is reaching the final stages of conversion from a coordinate based approach to the preset gesture approach.The user is now able to interact with the application quickly and efficiently by selecting and confirming an emotion, and sending a preset integer to the server based on the emotion selected. This allows for an altogether easier and faster experience for the user by cutting down the time required to select the configuration

for the wall, at the expense of being able to customize exactly what shape the wall will make. We think that creating the app in this way will ultimately afford a better experience to the user.

Emotive

As of October 27th, the Emotive headset prototype has been able to interact with the server as intended, when given artificial inputs (as opposed to natural inputs generated from the headset itself). The Emotive interacts with a separate page on the server which is intended for communicating coordinates and dimensions of a rectangle, rather than a simple emotion present integer.

Grasshopper (Rhino)

As of October 21st, Grasshopper is able to communicate between the Server and various physical wall prototypes which have been created. Grasshopper currently only supports communicating emotion pre-sets with the new server, and communicating coordinates with the old server. However at present it has not been updated to communicate coordinate data with the new server, and furthermore we do not have any physical wall prototypes which support this communication either.

Kinect

As of October 27th 2017, the kinect prototype has been updated to connect to the new server. Previously the kinect was sending the coordinates to the old server. Now the kinect has been updated so it sends its data to the same server as all of the other devices.

Server

Since the beginning of the semester, the server has been totally overhauled from scratch. We are now using a python server hosted on a new domain. As of October 25th our server supports both communications requested by the client, those being, emotion presets, and Coordinates/Dimensions respectively. The emotion presets have integer representation, whereas the Coordinates/Dimensions have a JSON Object representation which is comparable to the data-points that our original server handled. The server is able to handle inbound information update request for both of these, as well as outbound current information messages. Additionally the server is setup to log all of the communication for these two communication forms separately, and store this information into two separate databases. This database logging was requested as part of the next step which will be some form of machine learning based on the data collected.

Beta Prototype Test Results

The testing of our prototypes was done by bringing in users and having them test the prototypes with little to no assistance. The users were given the prototype and were told detailed instructions for how to operate each one. The users were then tasked with using the prototypes to manipulate a mocked wall. The results were that the users were able to manipulate the wall properly. Each device was able to take in user inputs and send that data to the server to move the wall. The only problem we faced was that users would have trouble remembering the methods used to input data. There were instances with the kinect and the alexa where the users could not remember the proper words or tended to forget which gesture corresponded to which action. The results were that the prototypes worked properly, but there is a need to make the means of operating each device more user friendly. This may mean that the prototypes will require

Beta Prototype Validation Results

The results were validated as we used a wide variety of users to make sure that any issues that needed to be addressed were ones that were produced by every type of user. The users ranged from male to female, young to old, and knowing of the technology to not knowing the technology. This wide range of people made it so that the team was able to verify that all potential users could properly use the devices. Since all of the users were able to properly manipulate the wall using our devices, it is safe to validate the results.

Broader Impacts Considerations

The concepts used to manipulate the wall also has the potential to help disabled people. Our project has been built around collecting data through devices located on and around a person. With this, we can expand into areas that require vast amount of data about the user. One instance of this could be assisting disabled people. With our data gathering, we could help disabled clients manipulate devices and objects that would normally be difficult to operate. This could allow people who normally live with an assistant to now live independently.

Summary of Work Remaining This Semester

Affectiva

The only work that is left on the affectiva is incorporating a wider range of emotions for the system. The Affectiva SDK provides a range of possible emotions that we can use, so the task largely falls on deciding which emotions we should track and which should be sent to GrassHopper. This may require further configurations in Rhino and GrassHopper but will not

require further development for the server, and only limited development for Affectiva. This should not take longer than two weeks to complete.

Alexa

We do not plan to add any more features to Alexa this semester. The only remaining work here is to clean up the code for the Alexa skill and to fix the bug with receiving invalid inputs.

Android App

The Android Application is nearing the end of its development cycle. As of right now, we have some plans to improve its capability and to increase the ability the user has to customize their experience. We plan to add compounded emotions as the final feature for the app. This means that a user may select an emotion such as “sad” and will be prompted with a slider to determine if the user feels only sad, sad and angry, or sad and scared. There will need to be custom configurations set in Rhino and GrassHopper to accommodate these changes. Our server is pre-built to handle these changes, so the only work that will have to be done is to configure Rhino and add a new screen to the Android app.

Grasshopper (Rhino)

Grasshopper needs to be updated to support coordinate communication with the new server. Additionally it must be updated to support outbound communication of coordinate data to physical wall prototypes.

Emotive

Testing needs to be done with the headset, and the Affective Machine learning must be taught on multiple users in order to provide a larger sample size for what brain activity corresponds to what activity. So far it has been shown to function properly with artificial manufactured data so our next step is using the naturally created data from the headset in order to communicate coordinate actions to the server directly from the user's thoughts.

Kinect

As for the kinect, we must now adjust the gestures so that a user can either change the wall to preset coordinates or send the data in themselves. There is also a need to improve the user interface to make the gestures easy to remember and quick to use. Currently there are too many complex gestures making it hard for users to manipulate the wall. There is also a need to update the backend of the kinect so that it will send coordinates to the new server. The kinect currently sends the data to the new server, but only for preset coordinates. The client has requested that the kinect allows the user to send both coordinates and preset configurations so the kinect must be updated to incorporate both. As of now, the remaining tasks will be spread across 4 weeks. The

first 2 weeks will be for smoothing out the kinect and have it send coordinates to the new server. The last 2 weeks will be for creating/revamping gestures to make them more user friendly.

Server

The server will need to be updated once we start to incorporate our machine learning algorithm into the system. In order for our server to learn the user's patterns, it must keep track of when each integer is sent to the server using a timestamp. We also may need to update our SQL database to better match the needs of our algorithm by adding more indexes or tables.

Appendix

Image 1: User interface for Android App