*The Many Faces of Microbial Communities*

Progress Report

Thomas Albertine & Michael Phelps

Electrical Engineering and Computer Science

Oregon State University

CS 461: Fall, 2015

*This document covers the 8 week process of research and planning for the senior capstone project between October 11th and December 4th. It will go over the various documents created and a timeline of the work which was done.*

# Introduction

In this document, we will go over the various tasks our group performed during the first 8 weeks of the senior capstone project. The particular project assigned to us was “The Many Faces of Microbial Communities” project, and our client is Dr. Jenna Lang of the University of California, Davis. The group members of the project are Thomas Albertine and Michael Phelps. The goal of the project is to create an application which can take in microbial sample data, and use the data to generate faces with attributes based on the data. The hope is that scientists will be able to compare the faces and use them to find patterns in the samples which they each represent. This is believed to be possible because the human brain has a portion specifically devoted to recognizing, distinguishing, and seeing similarities between different faces [1]. If this project is successful, it could produce huge breakthroughs for the field of microbiology and be expanded into other fields of study which require visualization and comparison of large data sets.

# Initial planning

## Initial planning of the Microbial community project was primarily done during the first week of the project between October 11th and October 15th. First, we met with the client over Google Hangouts for about an hour at the beginning of the week to get a better idea of what she wants out of the project to ensure that our work is properly in line with our client’s wishes. After meeting with the client, we began research on the various ways in which we could go about creating the visualization software which our client wanted. The client suggested that we use the MakeHuman open source software in order to generate our visualizations so additional time was spent analyzing the software and determining if it is appropriate for our project. We decided that MakeHuman seemed to be very suitable for our project and decided to move forward in our planning with the assumption that we would be using MakeHuman. We also discussed the prospect of creating a web based system as opposed to local software. It was decided that a web system would require additional work which may or may not be accomplishable within the time frame provided. We determined that the web based system could be put on our stretch goals for the event the rest of the software is completed faster than anticipated. It was decided that more specific aspects of the project could be decided at a later date.

## During the initial planning phase, we also created our problem statement. We had trouble fleshing out everything that needed to go into our problem statement and the document ended up being fairly poor. In the end, it we did not come out with a great problem statement but it had everything we wanted to to say on it and everything else was fleshed out more in the requirements document. A lot of our troubles were due to not properly understanding the assignment. The document ended up only being a page long and lacked a lot of the details which should have been included.

# Requirements Document

We spent weeks 2, 3 and 4 working on our requirements document between October 19th and November 6th. The document had to include a fully fleshed out list of requirements for out project which will be a basis for the grading of the project when it is completed. During week two we came up with a very simple overview of what should be required in our project. We came up with a list of general functionalities for the project and added user stories to better explain what should happen from that perspective. We decided not to use any specific format initially but later we determined that to be a pretty big mistake. Overall however, being a two person group we decided to keep our use cases and total requirement fairly limited while putting some possible functionality into stretch goals. This is also when we decided on our team name, ViewCrobe Software.

We struggled quite a bit at the point with what was actually supposed to go into our requirements document. We felt like it needed more content but we felt that our project was so one dimensional that it did not take a lot of detail to actually get everything that we needed. The typical use case is that a user selects files to use as sample data, selects options for the parameters and then has the program generate the faces and show them on the screen. Any variance from that is fairly small and generally was still covered within our document for the most part.

A lot of time was spent attempting to communicate with the client. The entire third week ended up being spent attempting to communicate with the client. An email was sent to the client on Monday of week 3, requesting approval and signature for the requirements document. We later discovered that the client was out of town that week and was unable to respond. We ended up acquiring an extension on the document due date until Wednesday of week 4. Due to the difficulties of communicating with the client and getting signatures for documents, it was decided that we should use an easier method of getting her signature for documents. We agreed to use HELLOSIGN, a service which allows users to easily apply legal signatures to documents.

At the beginning of week four it was revealed that the majority of the requirements documents which had been submitted were well below standard and that we had until Friday of that week to submit a rewrite. We decided to improve our requirements document using IEEE guidelines, even though we had not been graded, as our client had not signed the document yet. We ended up adding more details about the functionality of the software. We also added additional information for providing context such as the purpose and the scope of the project. We also added definitions in order to make it clear exactly what was being said at any given point in the document. This will hopefully prevent any misinterpretation of the document.

We also decided that we should have a stretch goals section to the requirements document. Throughout the whole process we have had trouble really figuring out how much we will be able to handle in the few months we have to complete this project. The Web UI, as mentioned earlier, was put into the stretch goals because neither of us have experience in creating web services from scratch. We also decided that we should put saving and loading visualization profiles into the stretch goals, as this would require a whole new set of scripts for generating, parsing, and reading profiles for the specific visualization settings, and is not necessary for basic functionality. Finally, even though it is fairly simple, we include the ability to export images as a stretch goal as it is not a particularly important part component of the project and the models are meant to be compared as they are through the GUI.

# Technology Review

The technology review ended up being rushed during the end up week 4. First we broke down what components we had to specify technology for and which options were going to analyze. These components ended up being the method of generating models, the programming language to be used, the GUI library, what file format to accept for sample data, and the file type to be used for storing the generated models.

The first thing we decided on was that we needed to review what is to be used to actually generate our model. We tried researching other methods of generating human or facial models other than MakeHuman but they all had inherent problems that could most likely not be overcome. The other options were really quite poor and it seems clear that without MakeHuman, this project would be completely doomed from the start because the only real option we would have had would be to create our own software from scratch to generate models. This would be a very difficult (and likely impossible) task for two college students to do with the span of a few month while still having the create the additional components of the project. Because of this fact, it is also clear the entire success of the project relies on our ability to manipulate the MakeHuman API to work within our own software, which does not seem to have been done by anyone else as far we can tell through our research.

We also had to figure out what programming language to use. It was decided that the whole system should be done using a single language in order to minimize complexity of having to communicate between different components. The language selection came down to what the group already knew and what preferences we had. This led us to concluding that Python [2] would be the best language for us especially because the MakeHuman software is already in python which should make it easier to include. Both group members are already familiar with Python and it is also a scripting language which makes it more suitable for parsing our data files and using the data to generate models.

Based on that decision, we had to decide on a UI library to use which is available in Python. We decided on Qt because it is popular and seems reliable. The UI library we chose really was not of huge importance because we are not creating a very complex UI. We also spent very little time on deciding what type of object file we wanted to generate. The decision essentially came down to what MakeHuman could output and what we could easily output to the screen. This led us to deciding that .obj would be the most practical file format to use.

One major issue we ran into when creating the technology review was the decision on what data format we would accept for the microbial sample data. Our client had previously given us a website [3] which has information on the various data formats used to store microbial sample data, so we spent some time going over the various formats and figuring out which one we liked the best. We ended up picking one similar to JSON but after emailing the client, discovered that she wanted our software to be compatible with a CSV-like format. Because all the formats are fairly similar and they all require minimal effort to parse, we decided not to fight our client’s wishes and modify our technology review accordingly.

# Poster Draft

The initial draft of our expo poster was done during week 5. November 9th through November 13th. We decided to take the template poster provided and simply make modifications on top of it. We were able to come up with some preliminary text describing the project but pretty much had to guess at what should be discussed. We had some problems actually getting content for the poster, so we simply left labelled, but otherwise blank areas where we planned on adding example screenshots of our finished software. We also had trouble coming up with what to put down in our conclusion section so we simply left that with some Lorem Ipsum text. We ended up also taking our team picture to be used on the poster during this time. We had some discussion on whether or not formal attire should be worn in the picture but after reviewing past years posters decided that it did not really matter either way.

After a brief chat with our TA on the Monday after week 5, we realized that we needed to make a few modifications to the poster. There were a few sections that needed to be modified and the conclusion section needed real content. It was suggested that we simply put our expected and/or desired results in the conclusion section.

Our finished poster ended up being formatted in a fairly simplistic maner. We put general information like the problem statement and solution, along with pictures of our software and a description of features in the middle. We decided that more technical information, such as the technologies we used and an overview of the technical approach which was taken, should be on the left. That way it is there for more technical audiences to see but not the center of attention and not in the way of people who have no use for the information. On the right we had our team picture and names of everyone involved, as well as the results section which is currently an overview of our desired outcome for the project.

The final version of our poster will of course be much more polished and will actually have the images in addition to labels. We may end up needing to make changes to the poster however, in the event that the space we set out for specific images is not enough, or too little, or happens to not be formatted correctly. Because of this, we will need to constantly re-assess the layout of the poster next term, based on how the project is looking. It seems very likely that our vision of the final project will expand greatly if we do not run into any major stumbling blocks and are able to accomplish the primary components in a timely fashion.

# Design Document

The design document was created during weeks 6, 7 and 8. Per the requirements for the assignment, the design document was created using the IEEE Std 1016-2009 standard. The design document has essentially all of the information from our previous documents. It includes an overview of the problem based on the problem statement which we created earlier in the term. It also includes what technology is used and what the software will be able to do which was determined in order to completely fulfill our requirements which we outlined in the requirements document.

The user-oriented views section of the document went over the use cases for the software along with a mockup for how the UI will end up looking. The use cases were essentially generated based on the use requirements from our requirements document. We outlined that our software will visualize data with the ability to configure parameters for the visualization. In the UI section, we went into more specifics about how each step of the process for taking sample data files and generating visual models would work. We also added some basic mockups of what the UI will end up looking like. Currently they are fairly simplistic in terms of styling but almost fully fleshed out in terms of functionality.

The developer oriented section went over both the technologies used and the more specific design aspects of the software project. The general structure of the software essentially follows the primary use case, of the user loading files, sending data to our translator script, which then goes through the model generator, and then onto the actual display of the model. The only complexity comes from the component which allows the user to modify parameters of the file. It is important to note that the data-model parameter section of our project could end up being a notably difficult section to overcome even though it needs to be done in order for our software to have any real flexibility controlled by the user.

A copy of the initial draft of the design document was sent to Nels at the end of week 7. Nels responded with some fairly minor suggestions at the beginning of week 8 and the appropriate modifications were made to the document. Due to the heavily structured nature of the IEEE standard, the document was fairly straightforward, although somewhat tedious.

# Current Status

During the first eight weeks of working on the project, a great deal of documentation was created which looks very promising going into winter term. We have created nearly all of the necessary documents needed for our project and can begin working on the software whenever possible. We are concerned that we do not even have preliminary code generated which we were told should have been accomplished by the end of the term. Additionally, our struggles with client communication are a worrying trend which could cause us problems if it continues. Luckily, our client is not technical and has no interest in micromanaging the way we create our software, which means there should be little need for constant communication during the next few months of the project.

At this point in our project, being fully committed to MakeHuman and having no alternatives, it is likely that our biggest struggle will be being able to use the API with our own software. Research was done earlier in the term to see if other people had accomplished what we wanted and the results were not very helpful. Many forum posts had been made and it seems that the API is not entirely meant for our task, but we are still trying to generate models based on our own parameters for individual facial features. If MakeHuman proves to be easy to work with, then the project could end up being completed within a fairly short time frame which would allow us to work on some of our stretch goals. If MakeHuman proves difficult to work with, then we could end up using a great deal of resources to figure a way to manipulate it to work with our code at all.

Overall, all of the ground work has been laid out over the past 8 weeks which should make it fairly easy going forward. Everything that needs to get has been fleshed out and everything that we are using to accomplish our tasks has been decided. Our UIs have also already been mocked up which leaves nothing left but implementation of the planning which has been done thus far.

# Citations

|  |  |
| --- | --- |
| [1] | "MakeHuman," [Online]. Available: https://bitbucket.org/MakeHuman/. [Accessed 3 November 2015]. |
| [2] | "Python," [Online]. Available: https://www.python.org/. [Accessed 4 November 2015]. |
| [3] | "Data Formats," [Online]. Available: http://www.metagenassist.ca/METAGENassist/faces/Docs/Format.jsp. [Accessed 6 November 2015]. |