Project 6 Report
By: Hannah Faus

Throughout the course of the semester, I was tasked with creating a chatbot in the programming language of my choosing. The chatbot was split into projects which focused on different specifications that the chatbot was expected to meet. Projects 1-5 were separate functions of the chatbot (with some overlapping) and in project 6, the previous projects were combined into a singular chatbot program. Project 6 consisted of the entire chatbot program as well as a report on the project and a presentation.

I chose to program the chatbot in C++ because that was a language I was unfamiliar with and it was the one we were being taught throughout the semester. I created a chatbot program that takes in a user's utterance and returns an appropriate response. One specification that I made is that the program reads in from a downloaded .txt file instead of directly from the browser. This means that if the webpage is updated, a new file will need to be downloaded to get accurate answers. Another specification that I made is that the buzzword and the answers are stored in the same file with a ":" separating them. This allows the buzzwords to be updated to make the chabot more accurate or to be changed if the chatbot becomes a learning chatbot. While those are the big specifications I made, a memory related one that I made is that I used vectors to store my information. This made it easy to access the stored information whether it was strings or ints!

The code was composed of a main function which called the functions that were requested. Depending on whether a "-" was included at the beginning of the user, utterance decided whether commandSearch() (for "-") or questionMatch() (w/o "-") was called. commandSearch() and questionMatch() were used to match the utterance to the correct response. commandSearch() would call sessionSearch() if it was able to find a command. Either way, createSession() was used to save the details of the user-program interaction. I tested all of the commands listed for each project as well as these commands with missing parts (i.e incomplete, misspellings, etc.) in order to anticipate possible user errors. The largest problem that I faced was data not being accessible in different functions. To counter this, I created global variables so that all of my functions could access and edit the data dynamically. Besides that, most of my problems were due to a lack of knowledge which I fixed from observing the lectures and reading public C++ code.

In order to make my code reusable I looked at the other senator profiles when creating the program aspects that involved matching user utterances to responses. For instance, some of the senator's cell phone numbers were labeled as either "cell", "mobile", or "home". I took specifications like this into account when making my program. In order to reuse my code all you would need to do is download the senator web page as a .txt file and place that into the data file and make sure the .txt file names matched the code and let the program run from there. Some challenges that I faced were that I wasn't able to get my code to read directly from a browser link, which would have made reuse easier, which I believe was because I was using a coding

language that was unfamiliar to me. I did not use anyone else's code because it was difficult to find other users who also programmed in C++ whose code made sense to me.

In the future I would first find a way to make my chatbot read the webpage dynamically so that when updates are made to the page (for instance: a new senator is elected or the senator moves) that information would be available to the user. Then, I would want to make my chatbot able to learn. So, when a certain number of users ask a question the chatbot can anticipate answers better or when the user asks something that the chatbot doesn't know, it can better interpret the correct response. Over time, my code would most likely become more efficient due to my growth as a programmer and new ways to implement things. I know this because I can already see more efficient ways to construct my earlier projects from the semester!