Gary Chen

Michael Abraham

Andre Hance

12/5/2021

CS 201

Technical Report

Our project has made significant progress since the start of the assignment and we have been able to implement a lot of different aspects needed for our project. Our project requires coding and writing from each group member, so it has been a process of dividing our work and getting more done. Michael worked on the python code mainly, Andre the bridge for the code mainly, and Gary worked on the Prolog code. Gary has been working on the presentation as well as the final project report, while Michael has been working on this technical report and editing the presentation and final project report. Andre has been working on making the code to make sure that it is efficient and is able to complete its task. When looking deeper into the different work we have done, there is a lot that went into the project.

Gary took the lead in creating our project by presenting the idea as well as being the first to implement his code. The idea was simple, but he brought up using a translator in order to complicate the project more and make it more challenging to do. After that Gary really got to work in implementing the code in Prolog. The Prolog code was a matter of creating classes and parent classes, and assigning the different variables to these factors. The variables in this case were the characters from the game Overwatch. We used the characters as the characters in this game have different variables that differ from other characters, but are also shared with other characters, essentially putting them in a group. Being able to put them in a group allowed us to characterize them and make databases with them. After creating the Prolog code Gary made sure to go above and beyond and start working on different aspects of the project that we due later down the road. Gary made sure to set the guidelines for what we will be doing next and what the next steps are.

Michael had been making sure that the development of the project has been running smoothly and making sure that deadlines were met and we weren't over working. The code for the project has been separated into three big parts which was making sure that the Python code, bridge code, and Prolog code all worked in tandem. This is still not something that we have been able to properly implement, but is something that we have been working towards, while making sure we are satisfied with the work we are putting out. Michael has been focusing on the smaller aspects like errors or inconsistencies in the code in order to make sure that the end result is uniform and organized. The Python code was essentially just copying the Prolog code's logic, but making it able to satisfy Python's conditions. Michael struggled with developing the code a bit due to its efficiency. In order to implement the code we had to work with parent classes, but the different factors that went into making the class became something that needed to be further looked at. Making the code was one part, but the other part was making the code able to be called which led to developments that aren't efficient, but meant for readability. The code also began being more about its ability to act in ways that allow for the user of the code to say what information they are looking for. After developing the code Michael went through the writing portion of the project adding to documents that had already been created and adding information that wasn't already there and editing for grammar.

Andre has been working on one of the most important parts of the project, that being the bridge. The bridge is the part of the code meant for running the entire program, so in order for us to successfully finish the project we would need the bridge to be completely developed. Andre had to do some research for developing the code since the example we had of bridges was more simple. The project itself was making a translator, and the implications of doing so is a fairly difficult concept. With Gary and Michael developing the coding parts, the bridge works with both parts and a new part entirely. Andre has had to go through the Python and Prolog code in order to develop a bridge that is able to run both programs. Moving forward with the code was strictly dependent on making sure Michael and Gary had their code developed, so it was also a matter of waiting for the two so they could sort through mistakes in the code that needed to be fixed.

Currently our code has three files developed, and the next steps are testing the code out. The Prolog code was developed first in the project and hasn't needed much adjusting. The one adjustment that was needed in the code was the specifying names to be more specific. The characters in the game have names that are unconventional, so Gary shortened them so they could be easily callable. However, Michael changed this to use official names for only some characters, so that the authenticity of the project would be better. It also allows research on said characters to be easily facilitated outside of the scope of the project.

The Python code was next to be developed and went through a few developmental changes. The initial creation of the python code was through using the method of class in order to generalize the characters as heroes and villains with more descriptive details being input as arguments. This required the development of specifying each character to their respective classes, but the process didn't simply end there. After that was the development of call-ability in the code. We made it that the user will ask what character is being looked for, and it is not possible to use variables in a class to a string input in order to make conditionals. This led to the creation of the dictionaries for each character meant for call-ability. By putting the different variables in the class into separate keys in a dictionary, the items were able to be called upon when using an input function. After discovering these factors it just became a process of copy pasting for each character that had already been listed in the Prolog code.

The bridge code is yet to be developed, but the idea would probably follow along the lines of previous examples of bridges used in other code. It would be a matter of compiling the code so that they are able to be run together and possibly organizing the specified outputs.

We believe that we can get the code done in a more timely manner as the development of the bridge code permits us to move further in development. Working on this project together has shown us that developing code is a matter of making sure that whatever you’re working on needs to get done in order for the team to move forward. We have been able to work together and support each other in aspects of the code that may have been difficult to develop.