Gary Chen

Michael Abraham

Andre Hance

Computer Science 201

12/11/2021

Final Progress Report

For our final project, we decided to implement a database that requests what the user is looking for and outputs that result. The motivation for this project was in line for the different options we were offered when it came to developing our final project. Out of the options presented to us, we figured that we could use the final project as an opportunity to challenge ourselves on how to use coding languages and methods that were discussed in class. This idea was the one that presented a fair amount of challenge as well as possible prospects for implementation. The code follows the Track 5 of the possible final projects, which is translation. Choosing this track for our final project was a good choice in terms of development as it was something that we had discussed not too long ago, meaning the idea was fresh in our minds. With all these factors in choosing this project Track as well as the project idea itself was the best course of action in developing a project that would work and wouldn't be impossible to develop with our coding knowledge. The idea for the project stemmed from brainstorming and choosing something that wasn't too specific and that could be used in many different developments.

The database contains information about characters from a game, that game being Overwatch. Overwatch is a first-person shooter game that comes from Blizzard Studios and is about characters with unique abilities battling against each other in order to complete objectives.

The game goes the extra mile to illustrate the lives and pasts of the characters in the game in order to give players a sense of empathy or just a simple understanding of these characters and why they fight. Each character in the game can be defined by a few different factors that make them unique outside of their customization. These factors are their sex, either male, female or characters who are robots, their affiliation, whether or not they are heroes, villains, or more neutral parties, and the role they play. These roles are divided into three categories; Dealers, characters who specialize in dealing damage to other characters, Supporters, characters who specialize in healing other characters, and Tanks, characters whose main purpose is to be the target of attacks. In addition to their names these defining traits allow characters to be differentiated from one another. This is useful because of the amount of characters in the game and that may be created as the game goes on.

From the release of the game Overwatch has had a diverse roster of characters that is only growing bigger and bigger as the game goes on. From having so many different characters there can only be some many traits that can be given to each character to make them unique. There are a few characters in the game that share the same traits as other characters which may lead to confusion if you know limited info about the game. This database is able to relay specific information about each character in the game and the different traits they possess. This is useful for learning about the different characters as well as organizing them in a manner that is efficient and easy to call for usability.

After the idea was conceptualized, we planned on the development of the code and what we were going to use in order to make the idea come to life. Being that we chose to do this using a translation we decided on the two programming languages we were going to use in order to translate one to another. We decided to use Prolog and Python for a few different reasons. Python

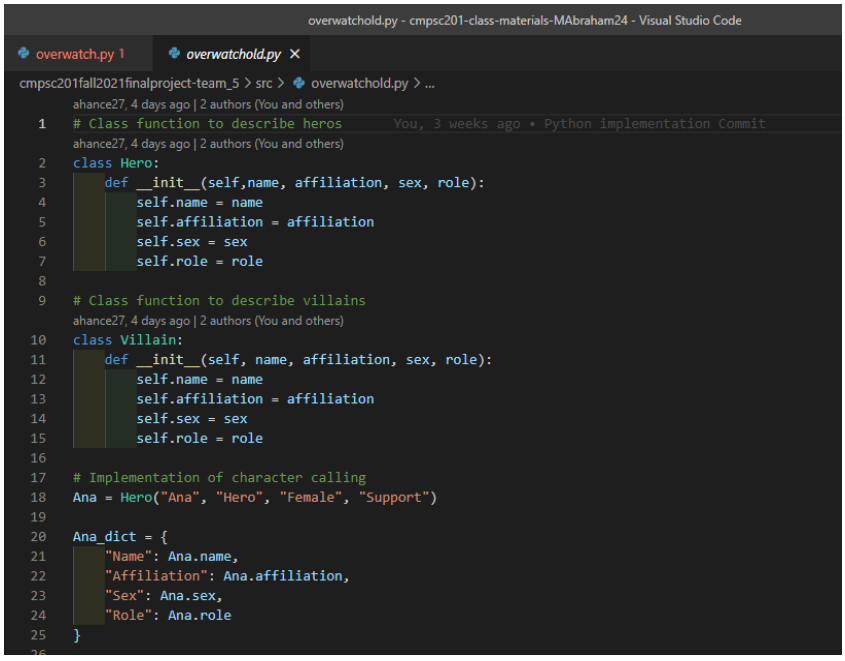
was one of the languages we chose to work with because of the amount of experience we all have with the language. We have been using Python as the main programming language in many other classes in the computer science department, so a language that we are more familiar with in terms of programming and logic made sense to use. We chose to use Prolog because of how recently we learned about it. The logic of Prolog was still fresh in our minds, so developing code with it would only take a bit of looking deeper into in order to develop. Prolog was also considered using based on the simplicity of the code. Prolog and Lisp could be noted as coding languages that are less human readable, but because they are less human readable they are also very useful in writing code due to the lack of long, descriptive functions. With those two established as the programming languages we were working with we also had to take in mind that we would need to develop code which would be used as the bridge in connecting these two programming languages together. Having the idea of what to do and how to do it was only part of the development process as we had to understand what we were developing.

Each of the programming languages were asking for something different in terms of development and in the first stage of our code we had made a big mistake. The Prolog code was developed first out of the bunch and it wasn't as bad as it seemed. Because it was a database it was a matter of cataloging information in a certain way and using the same method of cataloging the information for the different characters. After that was done came the development of the python code, which brought confusion. When developing the Python code the logic behind its development was that another database needed to be created which wasn't the case at all. After getting the code professionally looked at it was discovered that the Python code was to be used for user input and running the process in itself. After fixing this error and using professional consultation to develop the bridge came the output for the project. It was discovered that the

Prolog code was to be used for outputting the requested information of the user and thus came the development for that portion of the project. After it was developed the code was functional and we were able to move on from that portion of the project.

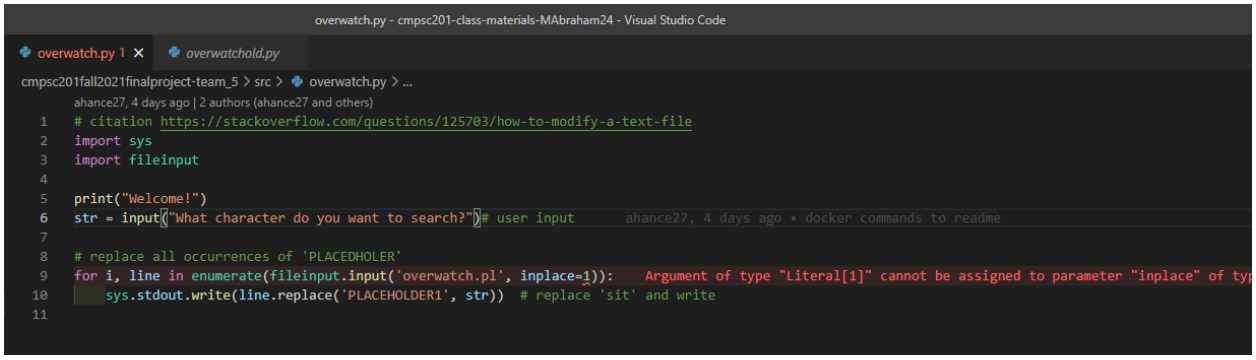
Developing the code was difficult because of our lack of information. Even though we learned a bit about Prolog and even with our knowledge of Python we still had difficulties. In relation to each other, what we thought we knew about the development of the Python code changed drastically. The logic of the Prolog code was also something that wasn't easily determinable either and the bridge between was a whole new subject. The development of the project seemed easy enough, but after starting it we understood that our assessment wasn't the case.

The Development of the Project:



For the python code this was the initial idea; creating classes for heroes and villains that had their specific traits used as arguments. The characters information would be input as a variable which would then be put into a dictionary in order to be called by the user.

This had to be changed because in terms of translation the code wasn't compatible with the Prolog code and was essentially useless.



After editing the code we were able to see that the actual development was much simpler and is able to fulfill its purpose. There appears to be an error, but it isn't detrimental to the code running and is simply because of a plugin for VS Code.

```
% defining male or female in overwatch
% split in both male and female
female(ana).
female(tracer).
female(widowmaker).
male(reinhardt).
male(winston).
robot(orisa).
robot(zenyatta).
male(cassidy).
female(dva).
robot(echo).
female(zarya).
male(baptiste).
female(mercy).
female(moria).        cheng2, a month ago • ad
male(lucio).
female(brigitte).
male(doomfist).
male(hanzo).
male(junkrat).
male(roadhog).
male(genji).
female(ashe).
female(mei).
male(reaper).
male(sigma).
male(soldier).
female(sombra).
male(ball).
female(symmetra).
male(torb).
female(phara).
robot(bastion).
```

```
% putting female into different groups hero vs villian
hero(ana).
hero(tracer).
hero(reinhardt).
hero(soldier).
hero(torb).
hero(echo).
hero(winston).
hero(genji).
hero(brigitte).
hero(zenyatta).
hero(phara).
hero(mercy).
hero(cassidy).
hero(lucio).
hero(dva).
hero(baptiste).
hero(mei).
villain(reaper).
villain(moria).
villain(junkrat).
villain(roadhog).
villain(sombra).
villain(sigma).
villain(doomfist).
villain(widowmaker).
villain(ashe).
villain(ball).
villain(bastion).
others(hanzo).
others(orisa).
others(symmetra).
others(zarya).
```

```
70  % defining roles in game.
71
72  sup(ana).
73  sup(moria).
74  sup(baptiste).
75  sup(brigitte).
76  sup(zenyatta).
77  sup(mercy).
78  sup(lucio).
79
80  dps(symmetra).
81  dps(junkrat).
82  dps(bastion).
83  dps(mcree).
84  dps(genji).
85  dps(hanzo).
86  dps(ashe).
87  dps(widowmaker).
88  dps(doomfist).
89  dps(sombra).
90  dps(reaper).
91  dps(mei).
92  dps(cassidy).
93  dps(phara).
94  dps(echo).
95  dps(torb).
96  dps(soldier).
97  dps(tracer).
98
99  tank(winston).
100 tank(sigma).
101 tank(ball).
102 tank(reinhardt).
103 tank(orisa).
```

Initially developing the Prolog code looked something like this, defining the roles and using parentheses to specify which characters are fulfilling these roles.

```
13  roles(hero, echo, dealer).
14
15  roles(villain, reaper, dealer).
16  roles(villain, moria, healer).
17  roles(villain, junkrat, dealer).
18  roles(villain, roadhog, tanker).
19  roles(villain, sombra, dealer).
20  roles(villain, sigma, tanker).
21  roles(villain, doomfist, dealer).
22  roles(villain, widowmaker, dealer).
23  roles(villain, ashe, dealer).
24
25  % seq(PLACEHOLDER1).
26
27
28  % relations and rules.
29  game(Organization, Name, Roles):-
30      roles(Organization, Name, Roles).
31      cheng2, a month ago • adding beg prolog file
32
33  main :- forall(game(PLACEHOLDER1, WhatHero, PLACEHOLDER2),(write(Who:game(PLACEHOLDER1, WhatHero, PLACEHOLDER2)), nl)).
34
35
36  % print character gender
37  % print role
38  % print organization
39
```

The revision of the code was made by allowing roles to be more broad which means that the length

of the code could be greatly reduced. The display statements were also added which allowed the code to be functional.

```
ahance27, 4 days ago | 1 author (ahance27)
1  python3 overwatch.py
2  swipl -s overwatch.pl
3  cat bkoverwatch.pl > overwatch.pl
4
```

Developing the bridge was much simpler than it seemed, but still needed to be researched due to the lack of information known about bridges

Working on this project has been extremely beneficial in learning more about how the code we have been working with works and opened our eyes to the process of developing more complex and elongated projects. We were able to learn how to divide up work and help each other in order to further the progress of the project and create something that we are proud of. Our project now completes its intended purpose and displays specified characters with their specific traits. The difficult challenge had to be the development of the Python code and the bridge because of the different factors we had to take into account when developing these two processes. From this challenge we now know how to develop something like this in the future and won't be as confused later on. The project has been helpful in furthering our experience and knowledge as programmers and has pushed us to do better in the field of computer science.