William Esposito

Professor Rivas

CMPT 220 Milestone Report

4 April 2017

Abstract

The following material corresponds to my Java application project. First, I will introduce the motivation of the work. After this will be a detailed system description, where I will explain how the program works. Directly after, I will cover requirements and also describe a similar system that I used as motivation for my own. Following that, will be a brief user manual and finally a conclusion.

Introduction

The New York Football Giants in the 2016 NFL season boasted an 11-5 record, and made the playoffs after a drought that went back to their super bowl victory in 2011. The purpose of my java application program serves as a way for a User to answer a series of questions with numbers, leading to more questions, eventually leading to the profiles of each starting player of the New York Football Giants.

Detailed System Description

The very first question asked to the user when the program runs is whether the player they are looking for is on Offense, Defense, or Special Teams. If the user wants to find a member on Offense, they will enter 1, 2 for Defense, or 3 for Special Teams. The number entered is stored as an int variable, which will determine which direction the program goes in next based on the number entered. This idea of the user entering a number, and being stored as an int variable which then navigates the program is something that is repeated throughout the code. Based on whether the user enters 1, 2, or 3, the program will then jump to a different block of code corresponding to the number entered.

Suppose the user is trying to find All-Pro Safety Landon Collins, when the first question is asked the user will enter “2” for defense. The number “2” is stored in the integer variable called “offenseOrDefence”. Now, int offenseorDefence = 2 since the user entered 2. The program detects this and a variable known as “directionNumberDefense” is increased by one. These “direction variables” are essential for displaying the correct text and guiding the program. Each position group as its corresponding variable, directionNumberOffense, directionNumberDefense, and directionNumberSpecialTeams.

So now, directionNumberDefense is the variable being used and navigating the program. It’s current value is 1, the program detects that and the block of code that corresponds is activated. A new question appears, “Please enter 1 for CornerBack, 2 for Safety, 3 for Linebacker, or 4 for Lineman.”. The number entered here is stored as a new integer variable, known as “defensePositon”. Depending on the value of defensePosition, directionNumberDefense is increased by a specific value corresponding to the position group the User is trying to find. Since we are trying to find Landon Collins, who is a Safety, we must enter the number “2” here. Since defensePositon == 2, the program will now add 2 to the variable directionNumberDefense. This is summated to the 1 that was already existing in the variable, now directionNumberDefense = 3.

This launches the block of code corresponding to the Safeties, because it satisfies the condition if(directionNumberDefense == 3). It reads, “You have selected Safeties. Enter 1 for Strong Safety. Enter 2 for Free Safety.”. Since we are looking for Landon Collins, we will enter “1”. This is stored in a new int variable simply called “safety” and since safety = 1, this satisfies the if(safety ==1) condition. Finally, the description of Landon Collins is printed, and the user is prompted to enter either 0 to exit, or 1 to continue and the program will restart and the user may find another player.

Requirements

For this program, any type of computer capable of compiling Java is required. A substantial amount of memory or computing power is not.

Literature Survey

One “game” that currently exists that can be loosely compared to my program is the Akinator game(http://en.akinator.com/). Akinator is a “web genie” that can take whatever real, or fictional character/person you are thinking of and based on how you answer certain questions he slowly narrows down his search engine until he finds the exact entity that you were looking for. He will ask questions like “Is your person real?” And depending on if you say yes, or no, certain parts of his database will be ignored for the rest of the search. He does this until he reaches the end of one of his search lines, and he will ask you if he got your person right. If you say yes, then the program ends and he has won. If he was incorrect, then he will ask a few more questions and give you an alternative guess.

The idea of answering questions and based off of the user feedback certain elements being displayed is the same, however my program is not nearly as vast and expanding as that of

Akinator’s. The Akinator will allow you to enter the character into the database if you indeed thought of something that he did not know; he grows smarter with every input. My program does not even use a database, and does not change based on if a user tries to locate a character that does not exist yet in my program.

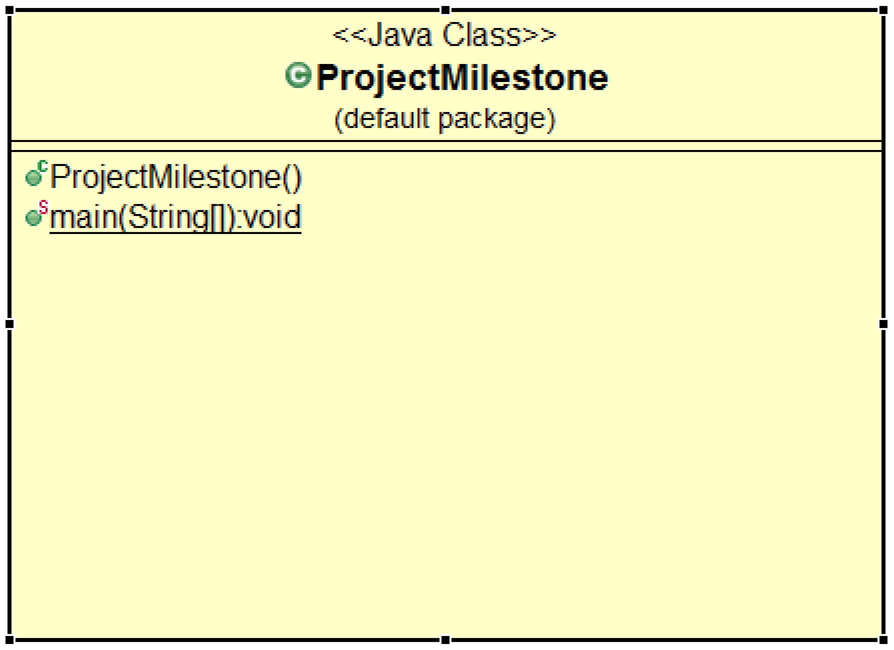
User Manual

- The first command for the user will be “"Please enter 1 for Offense, 2 for Defense, or 3 for Special Teams." And each command that follows from this one will follow a similar fashion in that you will have to enter a certain number corresponding to a certain player, or position. All you have to do is enter the number of who you are trying to find.

Conclusion

Any time a software developer is able to find something they are passionate about, but also able to apply their skills as a programmer there is an excellent opportunity to create a product they can be proud of. It is under such circumstances of my passion for the New York Football Giants, as well as my growing interest to program as a potential profession that this program was berthed and something I take pride in. Never before have I gone through the entire cycle of having an idea, working out how the program would work, and finally having a finished product that accomplishes what I originally sought out to do.

UML Diagram



<<Java Class>>
ProjectMilestone
(default package)
ProjectMilestone()
main(String[]):void