**Date: April 16, 2020**

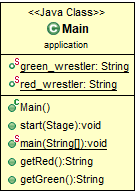
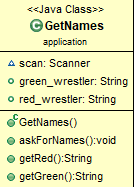
**To: Gilbert Munoz, PhD**

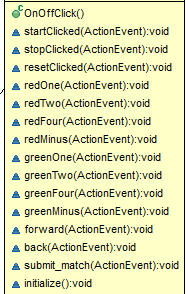
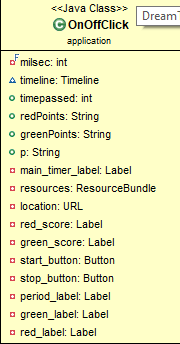
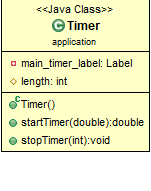
**From: Caleb J. Hopkins and Diavantae’ D. Reddick**

**RE: Final Project Report**

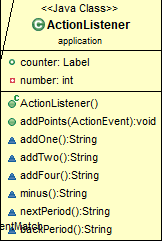
After a very successful semester learning the programming language Java under Dr. Munoz, Hopkins and I have put our heads together which led to the creation of a medium that will help in the officiating of wrestling matches on a grand scale. We have successfully developed an application that will serve the function of scoring wrestling matches or tournaments on the collegiate level. The desire to build an application as such stems both the creators attachment to wrestling as they both have been competitors within the sport since each of their childhoods. The second desire is derived from the seeing the obvious weakness of competing tools on the market; they fail to sufficiently and easily serve their purpose. The creators of this program are certain that they can capitalize on these flaws and turn out a better product than what is being advertised as the marquee application. For example, “[LevelChanger](https://www.levelchanger.com/)” is the scoring application used predominately in the United States. We firmly stand by the belief that just because a product doing the best in the market, does not mean that improvements should be halt. The “[LevelChanger](https://www.levelchanger.com/)” program has proven to be inefficient and at times, inoperable. For instance, when using the application in an official collegiate match at Campbell University in 2019, the application failed to let the users know that they were under a trial period for the duration of three hundred matches, and it also failed to display a counter of how many total matches had been used in the trial period. This alone created chaos due to the trial period being reached in the middle of a college wrestling dual accompanied by a delay in the action in order to settle this technical issue. With this, the wrestlers involved in the match were forced to wait until the situation with the clock was resolved. A break in the action like this can be the difference maker between a win and a loss. A simple technical issue should not have such a huge impact on the outcome of a match. An additional limitation that the top program has is that, it is limited to the Android operating system and Apple iOS. Our program strives to deliver the highest quality and efficiency with limited errors on a multitude of operating systems. This program will score wrestling season of the folkstyle variety with ease no matter how new the user is to the sport of wrestling.

The development of our code came about when first presented with the prompt and requirements for the final project. It was Hopkins’ idea to go ahead and stick close to home with the idea that we utilize for our project. We deliberated and came up with the idea to have wrestling be our central focus for this project. Scoreboards are a giant factor in the tracking of wrestling events so we eventually decided to create a scoreboard for wrestling with all the tools that we have learned throughout the semester. The main portions of our code that we knew that we had to implement tied back to data for the wrestlers, a start and stop time clock, three periods for regulation wrestling along with overtime periods, and ways to log point scoring. This information was created in eclipse via scanners, multiple methods with proper accessor methods, and the utilization of advanced methods in an fxml document and JavaFX. The remaining portions of the code such as the riding time aspects were additional features that we felt that we could add and would expand upon in later improvements or upgrades to our project.

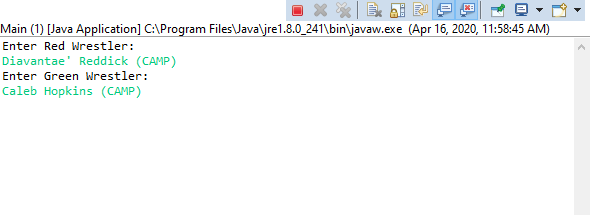
The structure of our program seems more complicated when first looked upon but after a deeper inspection of all the elements of our code, it abides to the fundamental practices of Java and is quite simple. Like every Java project created the Main is what allows the program to execute properly. The first two strings seen here “green\_wrestler” and “red\_wrestler” are names for the respective wrestlers involved in the match. These names are gathered from the scanner user input in the GetNames class.

An integral portion of this project comes from the timer. Without this part, you essential can not have a match.

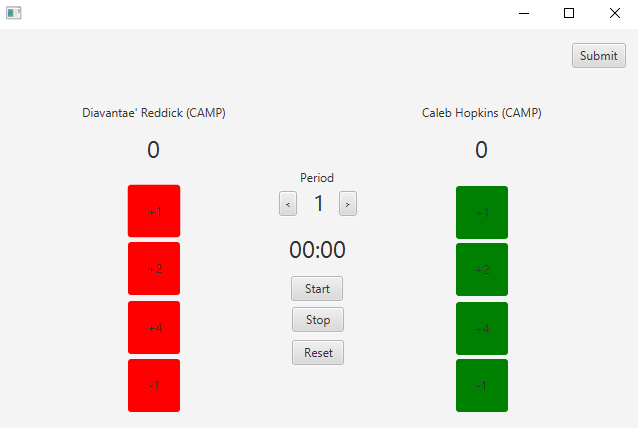
The timer class works hand-in-hand with the OnOffClick class to create a timer but also give it an interactive button the user may use to control the pace of the match for referee discretions, blood calls, injury timeouts, or out-of-bounds calls. The OnOffClick class then goes a step further to create a multitude of variables that create the buttons needed but also call for action events that tell the buttons what action to take when clicked which connects to our ActionListener class.

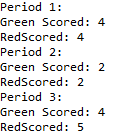


This class serves as a controller for the buttons. It gives each button a rule and affects the displayed value in JavaFX Without this class, when the buttons are clicked like +1, the rules would not tie to the button and it would never add one, it would just print 1 on the screen over and over again. The classes Match and CollegeMatch are additional files that tell each variable to add points to the the score under protected variables and sets the initial (0-0) score for each match and records the results of previous matches.

 Upon running the Java application, a prompt will emerge asking you to enter the names of the red wrestler and the green wrestler.

The magic really begins to happen after entering the names of both wrestlers. The user operating the Java application will be greeted with this beautiful screen:



Each button will print and add the respected point values depending on which wrestler achieves an escape, a takedown, and back points. A start and stop button to maintain the flow of the match. A reset button for when you need to reset the time when switching periods and if anything goes terribly wrong. A normal match will go three periods with the option to go into overtime periods if the scores are tied at the end of regulation. A submit button is also located in the top right corner that will submit the results period-by-period back into the console.



This is a great project because it made us realize that not only are we more than capable of producing a product that could be used on a grand scale but also it showed us the business side of it as well. An obvious lack of competitors in the market combined with our program being created with the weaknesses of competitors in mind gives us leverage to actually go through all the way if we wanted to. For now, this project will be kept close to us both so that we can focus on improving upon it so it can be the perfect tool for wrestling. These improvements include adding riding time for college matches, including a separate blood and injury time column, the possibility of adding a bell to signal the end of periods, and a migration to more operating systems.