**Project Documentation**

**Project Title: Football Tournament Simulation**

**1.1 Introduction:** A simple a Java software application that simulates a football tournament for sixteen (16) teams. Similar to that of a World Cup competition, it has Round of 16 up to the finals. Sixteen countries around the world have a team represented at the tournament. Only the second round to finals were considered. Each game was set up to have a winner, each team can either win or lose, via direct win or loss. All scores and points are randomly generated. There are also penalty rounds for when the same numbers are generated for both teams. The 8 teams with the highest goals, and penalty kicks if done, goes on to the 2nd round, and the best 4 to the semi-finals and the best two to the finals. The team that attains the highest score in the final is the winner.

**1.2 OutPut:**

Text

Description automatically generated**Fig 1.1**

Text

Description automatically generated

Text

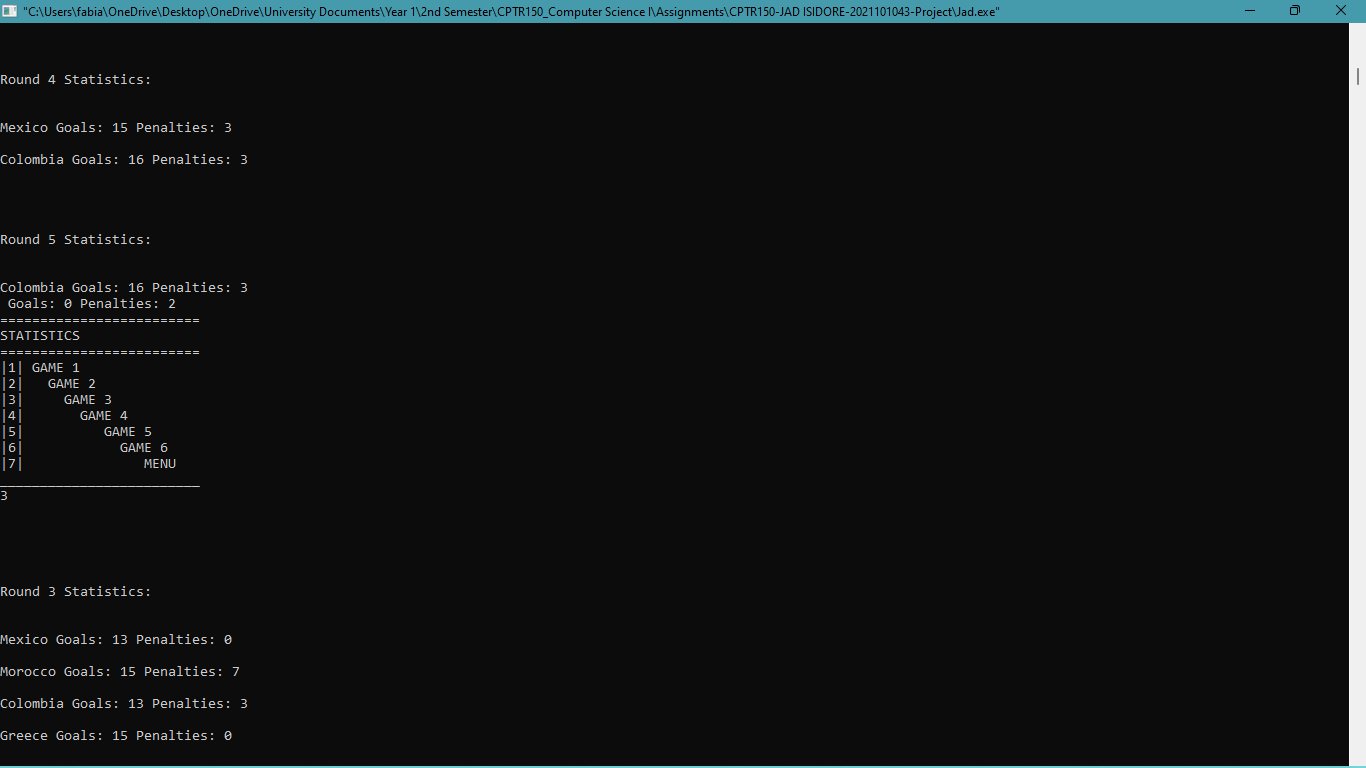
Description automatically generated

**Fig. 1.2**

Text

Description automatically generated

**Fig. 1.3**



**Fig. 1.4**

**2 Development:** My code's simulation of the tournament starts with the round of sixteen and goes all the way to round two. My text file, which identified all the teams competing in the tournament, the rounds that would be played to decide the victor, and the statistics that would show the list of teams and their goals at various stages in the competition, was displayed after I first displayed my introduction page.

**3. Coding Concepts:** Throughout the entire process, I extensively relied on strings, arrays, structs, do while loops, variables, and I used both global and scope variables. I started by breaking down the bigger issues into their component elements in order to make my task easier. In order to repeat the rounds for each of the several rounds, I created an array containing each of the rounds. Additionally, I created random numbers for each team using a random number generator. I also created numbers so that the team may win by penalties if the teams drew in the round. A struct name team was created bearing all the elements of any regular team, and was used to create arrays for the several rounds.

This was done in such a way that helper would complete these tasks within a particular range, in the fragments of code above, the ranges are 2 and 4. Moreover, the use of sequences played an integral role in the construction of this game as certain parts of the code were dependent on the rest. For example, the map first had to be customized, then after, the players, followed by the building of the items such as the arrows, walls and spikes. If the map had not created, the other parts of the game could not have been organized in a proper manner.

**Challenges:** The reason for many of the problems I had was that I didn't use the flow chart or the IPO chart when I first started working on the assignment. I went ahead and changed my IPO and flowchart after hearing what you had to say, and as a result, things began to make a little more sense to me. The only other difficulty I had while working on this subject was trying to finish the rounds in time for the competition. Due to this problem, I had to manually code each round one at a time in order to reach the desired outcome. In the end, I was successful in getting my code to produce every outcome I had hoped for. Although the experience was a little stressful, overall, it was a fantastic learning opportunity.