

Name: Rana Moezz Hassan

Course: CS 152 A

Date: 12/10/22

Title: Project 10- Simulating the knockout stages of the FIFA World Cup

Abstract:

- This is a computer science project that simulates the knockout stages of the FIFA World Cup 2022 starting from the round of 16.
- One of the key concepts used for this project was classes and object-oriented programming. A class is a template definition of the methods and variables in a particular kind of object and Object-oriented programming (OOP) is a computer programming model that organizes software design around data and objects, rather than functions and logic. This idea is crucial to building successful large, complex, and actively updated or maintained programs.
- In the first part of the project, I developed a python module “world_cup_simulate.py” whose methods and attributes would be imported into the main file “visualize_simulation.py”. The most important in the python module is the get_winner function which returns the probability of the “home” team winning against the “away” team. In the second part, I developed the “visualize_simulation.py” file that utilizes that utilizes the module from the first part of the project to simulate and visualize the knockout stages of the world cup using zelle graphics and tkinter. The dataset used in this project was scraped from wikipedia using the pandas module.
- The user should run the “visualize_simulation.py” file from VS Code or alternatively, run the file from the command line with the filename being the only argument. The user should expect to see a tkinter window with two buttons: “Open Window” and “Simulate”. The “Open Window” button opens up the zelle graphics window with empty boxes. “The Simulate” button simulates one round of fixtures and fills in the empty boxes with the winners.

Project Design Sketch/ Image:

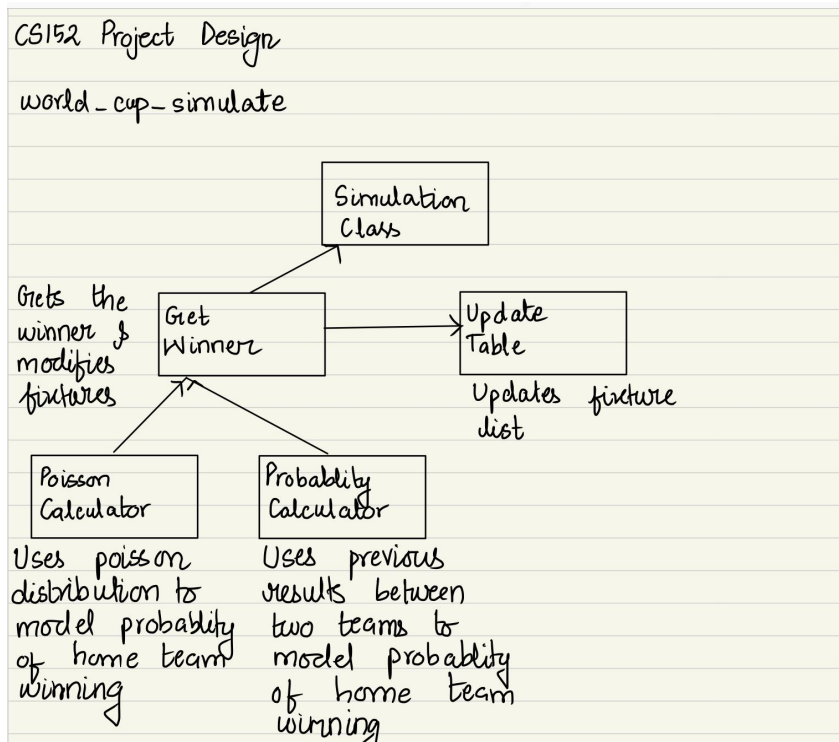
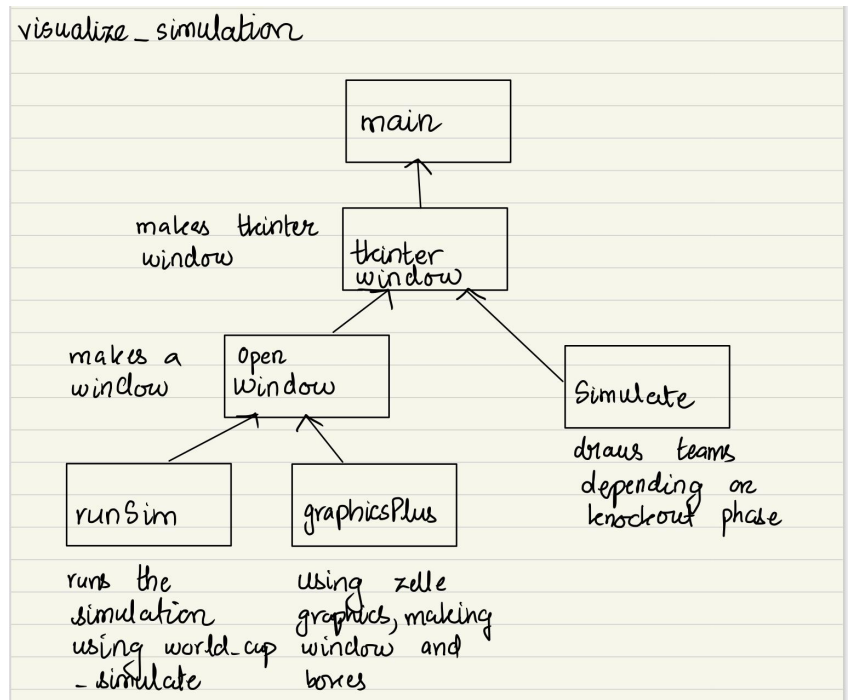


FIGURE 1: Project Design showing the hierarchy of the functions used in `world_cup_simulate.py`

FIGURE 2: Project Design showing the hierarchy of the functions used in `visualize_simulation.py`



Critical Evaluation:

If I would work on this project again, I would make several changes. I would use more recent and comprehensive historical data to train the predictive model. This would include data from the previous World Cup and other international competitions, as well as data on the teams and players involved in the current World Cup. Second, I would incorporate additional sources of information and analysis into the predictive model. This could include data on the playing conditions and weather at the World Cup. Third, I would update my fixture list in a different way. Instead of updating all of the values with one function, I would break it down into smaller functions that update the fixture list in small increments so that it would be easier to debug the program while developing the module. Lastly, the Poisson distribution is only appropriate for modeling situations where the number of occurrences is large, but the probability of any individual occurrence is small. This means that it cannot be used to model situations where the number of occurrences is small or the probability of an individual occurrence is large. Therefore, it might not have been the most appropriate distribution to model the probabilities.

Potential Biases/ Limitations:

The biggest problem with the results of this project would be the dataset that I was working with. Since I used only the historical data of the teams in the world cup, I ran into two problems with this method. i) when the two teams facing off against each other had never played against each other. I would have to then estimate the strength of the team based off their goals scored and goals conceded alone to decide the probability of one of the teams winning. ii) The World Cup is a unique event that occurs only every four years, the historical data used to predict the winner may not accurately represent the current state of the teams and players, the World Cup is a global event that involves teams from different regions and countries, and the quality of the teams and players may vary significantly and finally, the World Cup is a highly competitive and unpredictable event, and the outcome of each match can be influenced by a variety of factors such as the weather, the playing conditions, and the performance of individual players. This means that the historical data used to predict the winner may not accurately capture the complexity and uncertainty of the World Cup.

Results:

Please refer to the `visualize_simulation.py` and `visualize_simulation.mov` files in order to simulate and visualize the simulation of the knockout stages of the world cup. You will see a tkinter window with two buttons: “Open Window” and “Simulate”. The “Open Window” button opens up the zelle graphics window with empty boxes. “The Simulate” button simulates one round of fixtures and fills in the empty boxes with the winners.

Extensions:

Extension 1: The original `visualize_simulation.py` file was modified in order to run the world cup simulation was run 1000 times using the `world_cup_simulate.py` module and the winners were stored in a dictionary and then using `matplotlib`, the winners were automatically as a bar graph of team name against the times that team won the world cup. Please refer to the `extension1.py` and Figure 3 for reference.

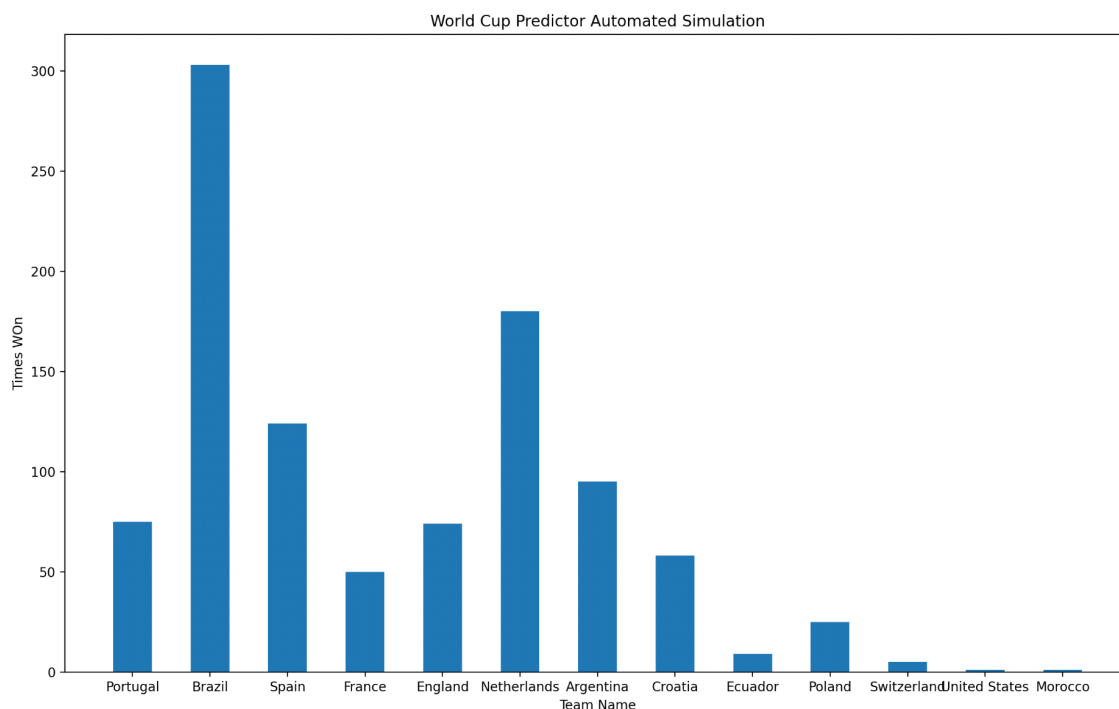


FIGURE 3: Bar graph showing the number of times a team won the world cup when the simulation was run 1000 times

I decided to do this extension to really understand just how biased my simulation program was and how accurate it is. This extension project helped me to deepen my understanding and mastery of this project, and to develop my critical thinking and problem-solving skills.

Reflection:

My project relied on the use of classes, lists and dictionaries to complete this project. Lists were used to store the schedule of matches and the results of each match and his information could be used to update the standings and rankings of the teams, and to determine the winner of the World Cup. This makes it easy to manipulate and process the data in a variety of ways, such as sorting, filtering, and transforming the data.

Classes were used to define and encapsulate the behaviour of the objects and behavior associated with the World Cup simulation and. This allows us to organize and manage complex data and behavior in a modular and reusable way, and to design and implement solutions to a wide range of problems in computer science.

Acknowledgments:

Professor Allen

Teaching Assistants

Frank Andrade - <https://www.youtube.com/@FrankAndrade5>

Peers