Design Document – EPL Score Predictor

## Description

Have you ever been talking about the weekend’s upcoming soccer games with friends, only for them to say: “Hey, what do you think the scores will be this weekend?” when in reality, you know nothing about soccer? Well, with this program, you never have to be in such an awkward position ever again! The English Premier League (EPL) Score Predictor can accurately anticipate any match-up within England’s top flight, using statistics from the 2014-2015 season (as of week 21), such as goals scored, wins, losses, and more. The program readily accepts any match-up and will display the score, with the team’s primary color on display with the text, as displayed in **Figure 1**. Another function of the EPL Score Predictor is that it can predict what an entire season will look like for any team; it simulates all 38 games at once, 19 at home, and 19 away from home, and then prints out the results and how many points the team would accumulate in total during the season. In general, this program is the ultimate score predictor, and with it, you can certainly impress your friends and show off to them even if you don’t know anything about soccer!

**Figure 1 – displays all of the Premier League’s teams**

As proof to the awesomeness of the EPL Score Predictor, the first image **(Figure 2)** provided displays what the EPL Score Predictor predicted for a game between Chelsea and Arsenal. In the second image provided **(Figure 3),** taken off Chelsea FC’s official website, you can see the actual result of this game which took place on October 5, 2014. Alas, the EPL Score Predictor was spot on! Although the program could randomly predict a different score line, the fact that it did show off a perfect score means that there is some legitimacy to it.

**Figure 2 – The EPL Score Predictor’s prediction**



**Figure 3 – The actual result between Chelsea and Arsenal; the EPL Score Predictor is right again!**

## Inputs

When the program is initiated, the user is prompted to answer whether they want a season generated, or a single game. If the user chooses the former, the program will ask which team’s season should be simulated. If the user chooses the latter, the program will ask for the home team and the away team who will be playing in the game.

## Classes Team:

|  |  |
| --- | --- |
| Field Name | Description |
| String teamName | The current team’s name |
| int gamesPlayed; | The total number of games played |
| int gamesHome; | The number of games played a t home |
| int winsHome; | The number of wins at home |
| int winsAway; | The number of wins away from home |
| int draws; | The number of draws |
| int lossesHome; | The number of losses at home |
| int lossesAway; | The number of losses away from home |
| int goalsHome; | The number of goals scored at home |
| int goalsAway; | The number of goals scored away from home |
| int concededHome; | The number of goals conceded at home |
| int concededAway; | The number of goals conceded away from home |

## Main:

|  |  |
| --- | --- |
| Field Name | Description |
| int numTeams | Set to 20, final. The number of total teams. |
| Team[] teams | The array that has to be filled up with all of the team names and stats. |
| int numHomeGoals | The number of home goals that are scored. Constantly changing. |
| int numAwayGoals | The number of away goals that are scored. Constantly changing. |
| Team home | A simple variable used to assist in getting user-inputs for the teams. |
| Team away | A simple variable used to assist in getting user-inputs for the teams. |
| String homeTeam | The home team string that the user inputs, which is then compared to the teamName variable to see which team the user has inputted. |
| String awayTeam | The away team string that the user inputs, which is then compared to the teamName variable to see which team the user has inputted. |
| String[] teamList | The 20 teams are inputted into this array, as they are read into from a file. |
| int[] stats | The 220 stats, 11 per team, are inputted into this array, as they read into a file. |
| boolean seasons | This variable is set to false until the user says that he wants a season (when prompted). After it is set to true, it helps generate an entire season. |
| Color[] colors | The array that has the primary color for each team (highlighted in **Figure 1**). |

|  |  |
| --- | --- |
| Method Name | Description |
| public static void calcHomeGoals(Team home, Team away) | The main formula for calculating the amount of goals the home team will score. Uses things such as goals scored, wins, losses, and more. |
| public static void calcAwayGoals(Team away, Team home) | The main formula for calculating the amount of goals the away team will score. Uses things such as goals scored, wins, losses, and more. |
| public static void drawPotential(Team home, Team away) | When the game is tied, but one team deserves to win (based on win/loss), this adds a goal to whoever deserves it. |
| public static void luckyGoals() | Sometimes, teams get lucky. This has a random chance to add a goal to either side. Slightly harder for home teams to get an extra goal this way. |
| private static void printStats(Team home, Team away) | Used this mainly to trouble shoot. Not a required method. |
| public static void realisiticResult(Team home, Team away) | This was an error filled method that was supposed to gift goals based on position in the league table; didn’t end up working though. |
| public static Team getHomeTeam() | This asks the user which team they want to use as their home team or for the season simulation. |
| public static Team getAwayTeam() | This asks the user the away team. |
| public static void getColors() | This sets each array value to a color, to match the teams. |
| public void paint(Graphics g) | Creates the field when the ‘game’ option is chosen, along with printing the score line on the field with colors. |
| public static String[] readInTeams() throws IOException | Reads in the team names from a file. |
| public static int[] readInStats() throws IOException | Reads in the team statistics (as of week 21) from a file. |
| public static Team[] createTeamsArray(String[] names, int[] stats) | This creates the main array of the Team object, with each Team having all the fields described from the team class. It uses the stats and names read in earlier. |
| public static void generateSingleTeamSeason(Team[] z) | This asks the user if they want a season or a game; if they choose a season, it will generate one for a team of their choice. Otherwise, it sets the variable **seasons** to false, which will trigger the if- statement in the main method. |
| public static void chelseaOP() | A fun little method that makes Chelsea a bit of a force to be reckoned with <:) Isn’t actually in use unless you put it in there though. |
| public static void main(String[] args) throws IOException | If the user wants a single game, the main method has the methods to formulate that game. Always runs generateSingleTeamSeason first, to see if the user wants that. |