NBA Player Guesser Game
Ojas Upalekar, Anish Jampana, Mehar Singh, Jaywoo Jo

Description

Our application is an NBA (National Basketball Association) game that allows fans to guess the player that satisfies a variety of given criteria, including aggregate player statistics, team wins, and boxscore season highs. By combining different types of stats together, as well as incorporating retired and current NBA players, our game is sure to be enjoyable for all fans who consider themselves NBA historians.

Usefulness

The goal of our application is to test the user's NBA knowledge. By providing a table of statistics, the user's ability to rationalize what player achieved the aforementioned statistics are tested.

A similar application to our project is an existing website game called "Poeltl". In Poeltl, users must guess a player in the NBA and are told whether the attributes of the player that they guessed are close or far from the player that was automatically selected. Some examples of attributes in "Poeltl" are "TEAM", "CONF", "DIV", "POS", "HT", "AGE", "Player Number".

Our game differentiates from "Poeltl" since we only provide statistics about that player's basketball performance, not physical attributes. Furthermore, "Poeltl" only includes current NBA players whereas our game includes all NBA players, allowing users to learn more about the history of the NBA.

Realness

Our data consists of player, team (including organization), and boxscore information from the National Basketball Association. Information will consist of statistics such as points, assists, rebounds, etc. for each player, as well as attributes that define teams, organizations and seasons. The data will come straight from the National Basketball Association's Developer API tool. Using nba_api, a package that easily connects the NBA's API will allow for easy lookup of data. Moreover, since we are using an API, this ensures that the data is dynamic and up-to-date as games are played.

Functionality

The website will function like an online quiz where users input the name that they believe is valid given the specific criteria. The user will begin by clicking a generate button so that the website pulls random pieces of criteria from aggregate player statistics, team wins, and boxscore season highs. The generated criteria will be the criteria that the user must guess for. A text bar, submit button, and give up button will appear on the screen once the user has clicked the generate button, and the user must continuously type in a name and submit. There is only one answer for the criteria and the user gets up to 10 tries to guess it. The score is based on how many guesses the user took to guess the player.

UI Mockup

i. Once the user begins the game, sample statistics for the player that the user has to guess will appear along with a space to enter his or her guesses. An example is shown below:

Guess The NBA Player
(given the following statistics)

Season	Team Wins	Aggregate Stats Example	Example Maximum Stat in that Season
2021-2022	33/82 wins	30.3 PPG (Points Per Game)	12 Assists (Twice vs. HOU and CLE)

You Have 5 *Guesses Remaining*

Enter Guess here

SUBMIT   GIVE UP

---

ii. If the answer was correct, a "Congrats!" pop up will appear. If incorrect, or the user pressed give up, a "Better luck next time!" pop up will appear instead.

Congrats!

The answer was LEBRON JAMES

Work Distribution

- Front-End (Ojas and Jay)
  - UI/UX Design of Landing page/Game page and overall framework. (Jay)
  - Counter for number of guesses (Jay)
  - Give up button (Jay)
  - Developing the main game table for given statistics and update on every user guess. (Ojas)
  - Create a pop-up on win/lose that allows the user to see more information about the player and the ability to play again. (Ojas)
  - Connection with back-end queries. (Ojas)
- Back-End (Anish and Mehar)
  - Create tables of Players, Team, Games, Box Scores etc. from NBA's Developer API. (Mehar)
  - Querying aggregate and max information for players. Finding max wins for a team. (Mehar)
  - Connecting queries with front-end. (Anish)
  - Log player guesses, wins, loses, history.
  - Be able to record and store number of guesses (Anish)
  - Compare guesses from user to list of players outputted from queries (Anish)