INST 326-01010
Team : Mint
Eunbi Kim, Vivian Hoang, Kiarash Torabazari, Colby Chung, Ahsan Khalid
Final project: football game
December 21, 2021

Problem to Solve

Kiarash Torabazari: The problem I set out to solve was to make a random.choice function that randomly pulls 2 teams from the list of teams I inputted and returns one from the AFC and NFC. Those will be the first 2 teams that are playing against each other. When running the first function it successfully returns 1 team from each list. I found the random.choice function to be the most efficient for this project. The user will then choose from the 2 randomized teams to play with.

Eunbi Kim: The problem I set out to solve was to read the text file for getting the roster of each team and find the team roster by using a specific team name. I make a class for the roster and then use two methods; one is to read the text file and the other is to find the team roster with its team name. Also, the problem I set out to solve was to calculate the team score and compare their score, then return who is the winner by identifying the score of the team.

Vivian Hoang: The problem I set out to solve was to generate a dictionary of both team's touchdowns and field goals. I found that the easiest way to do this was to use the dictionary method. A challenge I had to overcome was our topic as a whole since my knowledge about football is so limited so I had trouble choosing what aspect of the game I wanted to code for. Originally I wanted to write code that generated two graphs of the team's touchdowns and field goals so that I could check for correlation, but it was more complicated than I had anticipated. I overcame this challenge by choosing an easier method, but I plan on solving that original problem for my second function.

Colby Chung: The problem that I set out to solve was to create the teams class and to implement a quote on quote power level for each team. In the NFL each team has a different skill level compared to the other so I wanted to have that aspect in the program. Therefore, I create the teams class so that each team has a level of skill in terms of running the football and throwing the football. I used the random python library to generate a random skill level for each team. Before the game starts, the user can see their team's stats and increase the skill level of the running or throwing skill once. I also implemented part of the main method as well as the function in the main method to ask the user to increase the skill level.

Ahsan Khalid: The problem that I set out to solve is to create a class that reads the players and plays from the file and prints it. The other class is to read players from the txt file and can add the players to that file. Finally, the last class reads the plays from another file and shows the points/scores from each play.

Functionality

Eunbi(Roster.py):

- Input text file which has the roster of the teams
- Match regular expressions with team name
- Read the file to return dictionary
- Find team's roster with using the team name
- Return to show the team name and the team' roster

Kiarash(Football.py)

- Create two lists with teams consisting of the NFC and AFC
- Randomly choose between the 2 teams
- Return 2 teams (One from afc, one from nfc)
- Allow user to choose between 2 teams
- Raise an error "Sorry that is not an option" if the user inputs the wrong team or value needed.

Colby(Skills.py)

- Create the team object with running skill, throwing skill, skill tokens, and roster
- Each team has a throwing skill and running skill that goes up to 100
- Generate random int from 0 - 100 to determine throwing skill and running skill
- The function checks if the team has skill tokens and if so the skill token gets used up and the team skill will increase based on which skill the user wanted to increase.

Ahsan(read.py)

- Create three classes with one main function
- Class File has two methods, one for reading the file and printing it, other for closing the file
- Class Players has two methods, one for reading the players from a file and other for adding players to the file
- Class Plays has one method, to read and print the file. It reads from the file and has the points of each play.

Eunbi (score.py)

- Pick random numbers from 1 to 3 for even
- Pick random numbers from 1 to 3 for odd
- Calculate total score with adding even, odd, touchdown
- Return the total score with string

Project Architecture

Eunbi(roster) : Reading text file and identifying the team name and roster

Kiarash(footballteams.py) : Randomly choosing 2 teams from the AFC and NFC lists and allowing the user to input the team that they would like to use.

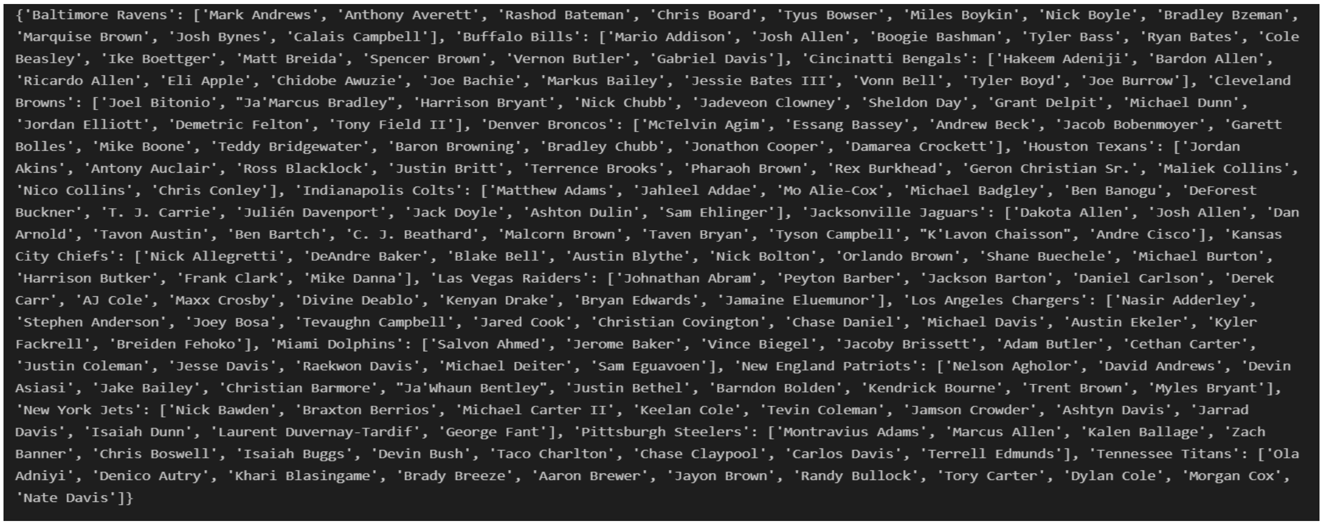
Colby(skills.py) : Initizes the team object and sets stats for the team.

Ahsan(read.py) : Reading the text file and identifying the players and the plays.

Eunbi(score.py) : Calculating total score for each team

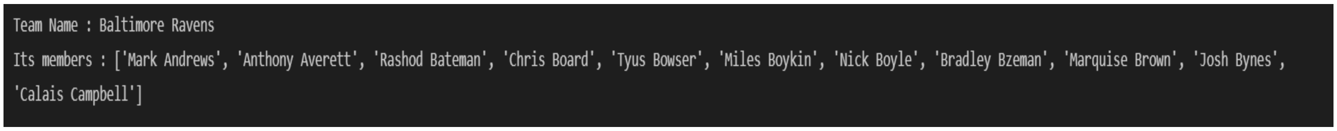
Expected Results

Eunbi(Roster.py) :
If using the contestants.txt,
The expected results of roster_file() would be:



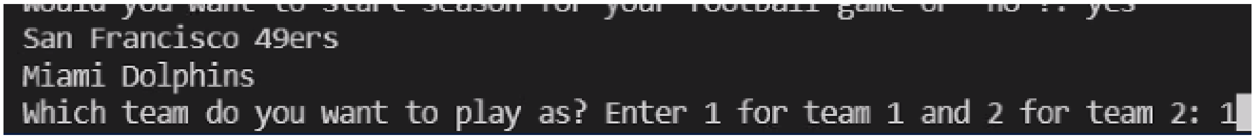
```
{'Baltimore Ravens': ['Mark Andrews', 'Anthony Averett', 'Rashod Bateman', 'Chris Board', 'Tyus Bowser', 'Miles Boykin', 'Nick Boyle', 'Bradley Bzeman',
'Marquise Brown', 'Josh Bynes', 'Calais Campbell'], 'Buffalo Bills': ['Mario Addison', 'Josh Allen', 'Boogie Bashman', 'Tyler Bass', 'Ryan Bates', 'Cole
Beasley', 'Ike Boettger', 'Matt Breida', 'Spencer Brown', 'Vernon Butler', 'Gabriel Davis'], 'Cincinatti Bengals': ['Hakeem Adeniji', 'Bardon Allen',
'Ricardo Allen', 'Eli Apple', 'Chidobe Awuzie', 'Joe Bachie', 'Markus Bailey', 'Jessie Bates III', 'Vonn Bell', 'Tyler Boyd', 'Joe Burrow'], 'Cleveland
Browns': ['Joel Bitonio', "Ja'Marcus Bradley", 'Harrison Bryant', 'Nick Chubb', 'Jadeveon Clowney', 'Sheldon Day', 'Grant Delpit', 'Michael Dunn',
'Jordan Elliott', 'Demetric Felton', 'Tony Field II'], 'Denver Broncos': ['McTelvin Agim', 'Essang Bassey', 'Andrew Beck', 'Jacob Bobenmoyer', 'Garett
Bolles', 'Mike Boone', 'Teddy Bridgewater', 'Baron Browning', 'Bradley Chubb', 'Jonathon Cooper', 'Damarea Crockett'], 'Houston Texans': ['Jordan
Akins', 'Antony Auclair', 'Ross Blacklock', 'Justin Britt', 'Terrence Brooks', 'Pharaoh Brown', 'Rex Burkhead', 'Geron Christian Sr.', 'Maliek Collins',
'Nico Collins', 'Chris Conley'], 'Indianapolis Colts': ['Matthew Adams', 'Jahleel Addae', 'Mo Alie-Cox', 'Michael Badgley', 'Ben Banogu', 'DeForest
Buckner', 'T. J. Carrie', 'Julién Davenport', 'Jack Doyle', 'Ashton Dulin', 'Sam Ehlinger'], 'Jacksonville Jaguars': ['Dakota Allen', 'Josh Allen', 'Dan
Arnold', 'Tavon Austin', 'Ben Bartch', 'C. J. Beathard', 'Malcorn Brown', 'Taven Bryan', 'Tyson Campbell', "K'Lavon Chaisson", 'Andre Cisco'], 'Kansas
City Chiefs': ['Nick Allegretti', 'DeAndre Baker', 'Blake Bell', 'Austin Blythe', 'Nick Bolton', 'Orlando Brown', 'Shane Buechele', 'Michael Burton',
'Harrison Butker', 'Frank Clark', 'Mike Danna'], 'Las Vegas Raiders': ['Johnathan Abram', 'Peyton Barber', 'Jackson Barton', 'Daniel Carlson', 'Derek
Carr', 'AJ Cole', 'Maxx Crosby', 'Divine Deablo', 'Kenyan Drake', 'Bryan Edwards', 'Jamaine Eluemunor'], 'Los Angeles Chargers': ['Nasir Adderley',
'Stephen Anderson', 'Joey Bosa', 'Tevaughn Campbell', 'Jared Cook', 'Christian Covington', 'Chase Daniel', 'Michael Davis', 'Austin Ekeler', 'Kyler
Fackrell', 'Breiden Fehoko'], 'Miami Dolphins': ['Salvon Ahmed', 'Jerome Baker', 'Vince Biegel', 'Jacoby Brissett', 'Adam Butler', 'Cethan Carter',
'Justin Coleman', 'Jesse Davis', 'Raekwon Davis', 'Michael Deiter', 'Sam Eguavoen'], 'New England Patriots': ['Nelson Agholor', 'David Andrews', 'Devin
Asiasi', 'Jake Bailey', 'Christian Barmore', "Ja'Whaun Bentley", 'Justin Bethel', 'Barndon Bolden', 'Kendrick Bourne', 'Trent Brown', 'Myles Bryant'],
'New York Jets': ['Nick Bawden', 'Braxton Berrios', 'Michael Carter II', 'Keelan Cole', 'Tevin Coleman', 'Jamson Crowder', 'Ashtyn Davis', 'Jarrad
Davis', 'Isaiah Dunn', 'Laurent Duvernay-Tardif', 'George Fant'], 'Pittsburgh Steelers': ['Montravius Adams', 'Marcus Allen', 'Kalen Ballage', 'Zach
Banner', 'Chris Boswell', 'Isaiah Buggs', 'Devin Bush', 'Taco Charlton', 'Chase Claypool', 'Carlos Davis', 'Terrell Edmunds'], 'Tennessee Titans': ['Ola
Adniyi', 'Denico Autry', 'Khari Blasingame', 'Brady Breeze', 'Aaron Brewer', 'Jayon Brown', 'Randy Bullock', 'Tory Carter', 'Dylan Cole', 'Morgan Cox',
'Nate Davis']}
```

The expected results of find_roster with that text file:



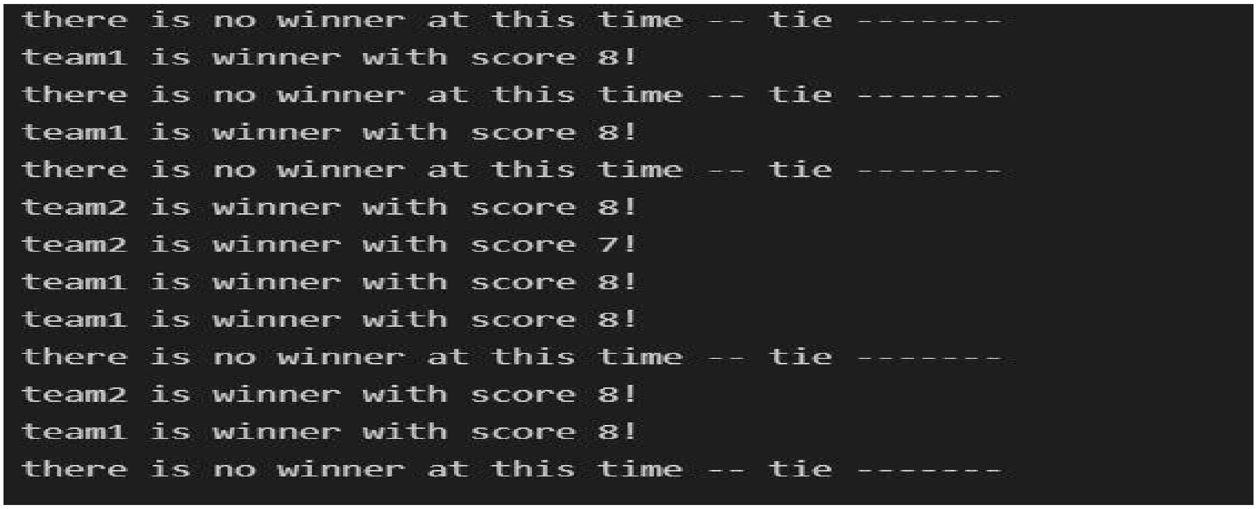
```
Team Name : Baltimore Ravens
Its members : ['Mark Andrews', 'Anthony Averett', 'Rashod Bateman', 'Chris Board', 'Tyus Bowser', 'Miles Boykin', 'Nick Boyle', 'Bradley Bzeman', 'Marquise Brown', 'Josh Bynes',
'Calais Campbell']
```

Kiarash(footballteams.py) :
The expected results of the footballteams.py



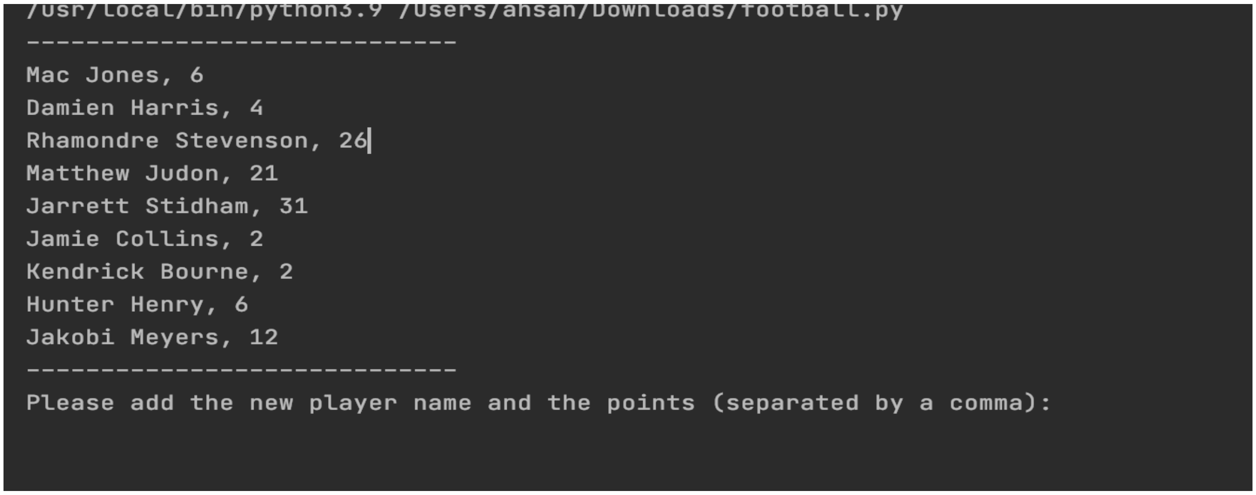
```
Would you want to start season for your football game or "no"?: yes
San Francisco 49ers
Miami Dolphins
Which team do you want to play as? Enter 1 for team 1 and 2 for team 2: 1
```

Eunbi(score.py):
The expected results of the score.py:



```
there is no winner at this time -- tie -------
team1 is winner with score 8!
there is no winner at this time -- tie -------
team1 is winner with score 8!
there is no winner at this time -- tie -------
team2 is winner with score 8!
team2 is winner with score 7!
team1 is winner with score 8!
team1 is winner with score 8!
there is no winner at this time -- tie -------
team2 is winner with score 8!
team1 is winner with score 8!
there is no winner at this time -- tie -------
```

Ahsan (read.py)



```
/usr/local/bin/python3.9 /Users/ahsan/Downloads/football.py
-----------------------------
Mac Jones, 6
Damien Harris, 4
Rhamondre Stevenson, 26
Matthew Judon, 21
Jarrett Stidham, 31
Jamie Collins, 2
Kendrick Bourne, 2
Hunter Henry, 6
Jakobi Meyers, 12
-----------------------------
Please add the new player name and the points (separated by a comma):
```

Challenge

Eunbi(Roster) : The most challenging is to get the data from each team's websites. There is lots of information. Also, organizing the data separately as a single text file is challenging to me because I need to check whether the text file has misspelling because if it has any misspelling, it would be shown error when I try to run the Roster.py.

Kiarash(Football): The most challenging part while creating my functions was deciding whether to use random.sample, random.shuffle, or random.choice. I also had problems when programming my code to choose between the 2 lists but I fixed that.

Colby(Skills.py): I think that the most challenging part while creating my function was trying to figure out how to implement the increased running skill and increase throwing skill method. Another part that I found challenging overall was to find time where all of the group members were available to work and to combine all of our parts into one program that works.

Ahsan(read.py): The most challenging part of my part was to link multiple txt files to my script and have my program read them in order to provide the output.

How to overcome the challenge

Eunbi Kim(Roster)**:** I try to match no more than 11 players for each football team. Also, I find the real roster from each team' website and get the data for the roster. Then, put the data into the text files so that there are two text files, one is roster_team and roster_contestants for a football match.

Kiarash(Football): After playing around with my code for quite a while I found that random.shuffle would be the best case scenario for the program that I was generating.

Colby(Skills.py) : To overcome my challenges, I just had to think about how I was going to make my methods logistically first then put it into code so that I had a blueprint of what I wanted to do.

Ahsan(read.py): To overcome my challenge, I made multiple classes in order to keep my script organized and have different classes to read the plays and players. The program also asks the user to add any new player names and score.

Actual Result

Eunbi Kim(Roster): My code gets the results of a dictionary of teams' names as keys and team's roster as values, and matches the regular expression successfully with team name. Also, if the user puts team name with the roster dictionary, it will return the team's roster as a list. But, if the team name is not in the roster dictionary, it will return ValueError.

Kiarash(Football): My code results in generating 2 random teams, and prompting the user to choose between the 2 teams generated. If the user inputs a wrong team then the program will say "Sorry that is not an option".

Colby(Skills.py): My code results in the creation of the Team object and the ability to increase the running and throwing skill of the team.

Ahsan(read.py): My code results in generating players and their score as provided in the txt file.

Eunbi(score.py): My code results to calculate the team score using random.randit() and adding them with the touchdown.

Reference

```
"""reference for
roster_team:
            https://www.azcardinals.com/team/players-roster/,
            https://www.atlantafalcons.com/team/players-roster/,
            https://www.panthers.com/team/players-roster/,
            https://www.dallascowboys.com/team/players-roster/,
            https://www.chicagobears.com/team/players-roster/,
            https://www.detroitlions.com/team/players-roster/,
            https://www.packers.com/team/players-roster/,
            https://www.giants.com/team/players-roster/,
            https://www.therams.com/team/players-roster/,
            https://www.vikings.com/team/players-roster/,
            https://www.neworleanssaints.com/team/players-roster/,
            https://www.philadelphiaeagles.com/team/players-roster/,
            https://www.49ers.com/team/players-roster/,
            https://www.seahawks.com/team/players-roster/,
            https://www.buccaneers.com/team/players-roster/,
            https://www.washingtonfootball.com/team/players-roster/"""


            """reference for roster_contestants:

            https://www.baltimoreravens.com/team/players-roster/,
            https://www.buffalobills.com/team/players-roster/,
            https://www.bengals.com/team/players-roster/,
            https://www.clevelandbrowns.com/team/players-roster/,
            https://www.denverbroncos.com/team/players-roster/,
            https://www.houstontexans.com/team/players-roster/,
            https://www.colts.com/team/players-roster/,
            https://www.jaguars.com/team/players-roster/,
            https://www.chiefs.com/team/players-roster/,
            https://www.raiders.com/team/players-roster/,
            https://www.chargers.com/team/players-roster/,
            https://www.miamidolphins.com/team/players-roster/,
            https://www.patriots.com/team/players-roster/,
            https://www.newyorkjets.com/team/players-roster/,
            https://www.steelers.com/team/players-roster/,
            https://www.tennesseetitans.com/team/players-roster/
```