Heath Aaron Lovell

ENGR 133-003

Individual Project Report

**DESCRIPTION**

**Main:**

This program allows the user to choose a team from the English Premier League and play a soccer game that is word based. First, the user chooses a team. Second, the user determines whether to attempt a ‘short’ or ‘long’ pass. The short pass is easier to complete with a lower pass value, while the long pass is harder to complete with a higher pass value. After attempting a pass, the user can either choose to pass again or to take a shot. The shot has a better probability of scoring if more passes are made. If the shot is missed, there is an opportunity of taking a corner kick.

**UDF – Pass:**

The pass function determines whether a pass is complete by obtaining a random number and checking it against a set probability. The short pass is more likely to be complete than the long pass. The function returns a lower value for a short pass than a long pass.

**UDF – Shoot:**

The shoot function determines whether a shot is made by obtaining a random number and checking it against a probability calculated by how many passes were complete. The function then returns a variable that tells the main program whether the shot was made.

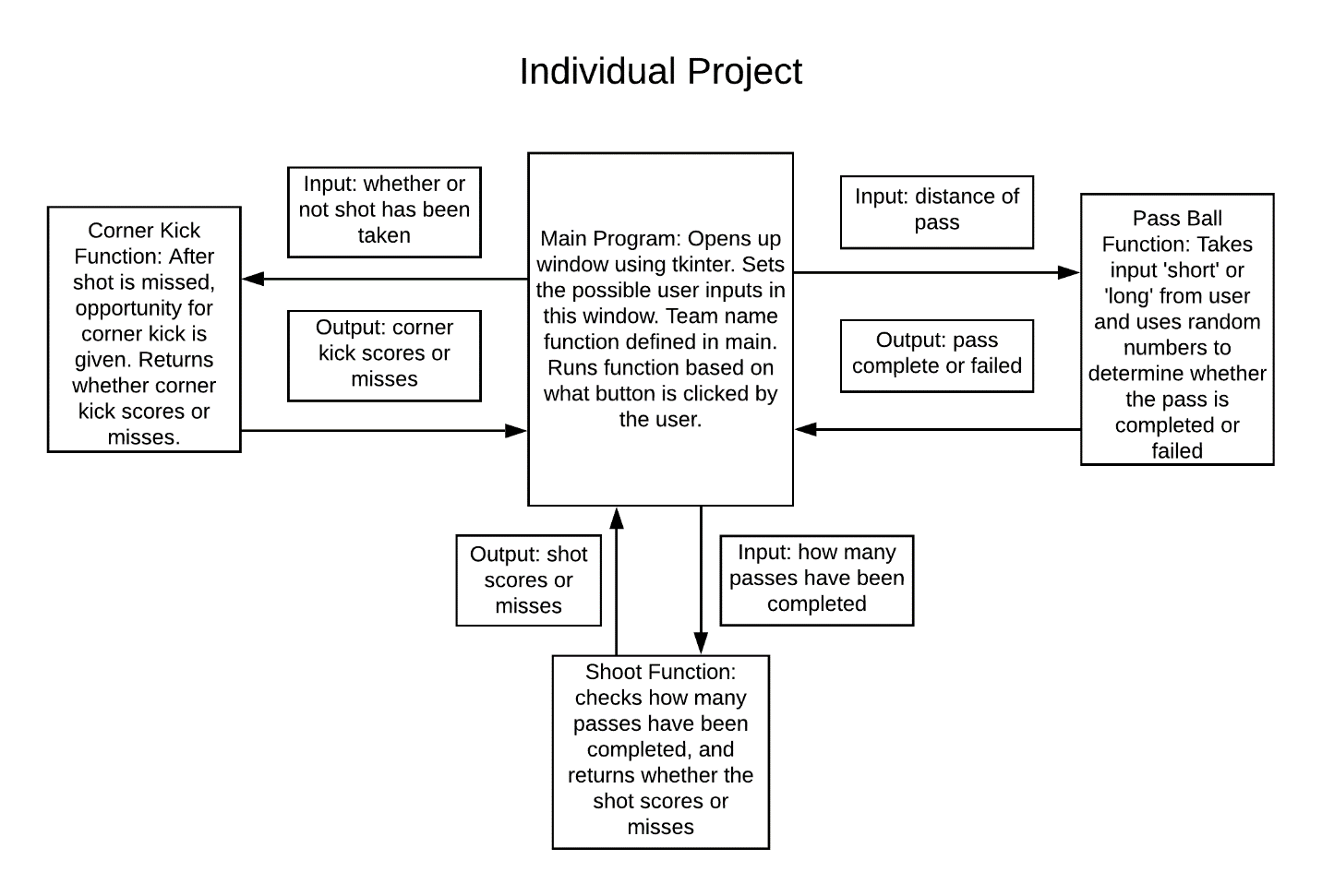
**UDF – Corner:**

The corner function determines whether a corner is made by obtaining a random number and checking it against a set probability. The function only allows the user to attempt a corner kick if a shot is not made. If the user makes a shot before attempting a corner, the corner function will tell the user that the shot was made.

**Line Numbers for Specifications:**

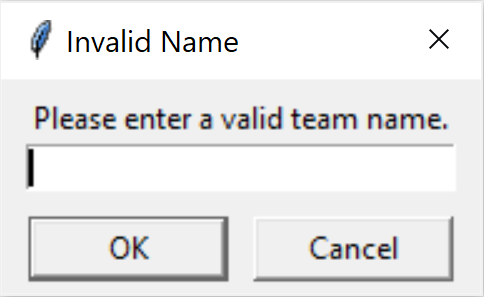
* User Input: **Main lines: 66, 81, 88**
* UDF Call #1: **Main line: 83**
* UDF Call #2: **Main line: 96**
* UDF Call #3: **Main line: 101**
* For loop: **Main line: 73**
* While loop: **Main line 87**
* Imbedded loops: **Main: While – line 70, for – line 73**
* Vector: **Main line: 85**
* If (with else of elif): **Main line: 67 & 69, 82 & 86**

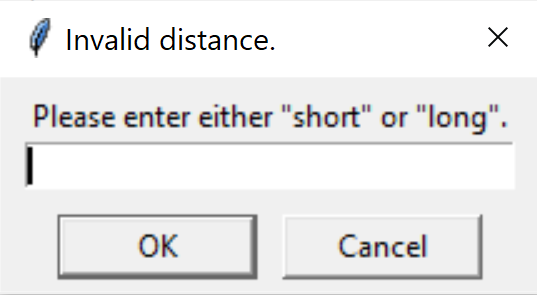
**INTERACTION DIAGRAM**

****

**USER MANUAL**

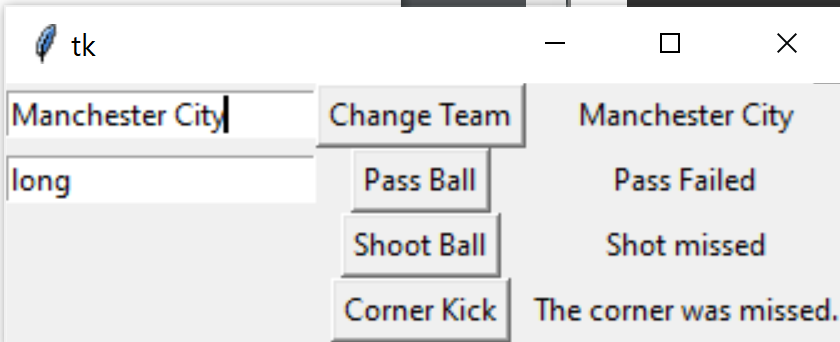
**Instructions for use:**

1. ****The user must enter the team name that they choose to play as. They must choose a team that is in the English Premier League. If an invalid team name is entered, the user will be prompted to enter a correct one as shown below.
2. After choosing a team, the user will then be able to pass the ball.
   1. The options for passing are ‘short’ and ‘long.’ A short pass is easy to complete with a low value, while a long pass is harder to complete with a high value.
   2. The values for the passes determine how easy the shot is to make.
   3. If the user enters an option that is not ‘short’ or ‘long,’ they will be prompted to enter a valid option as shown below.

****

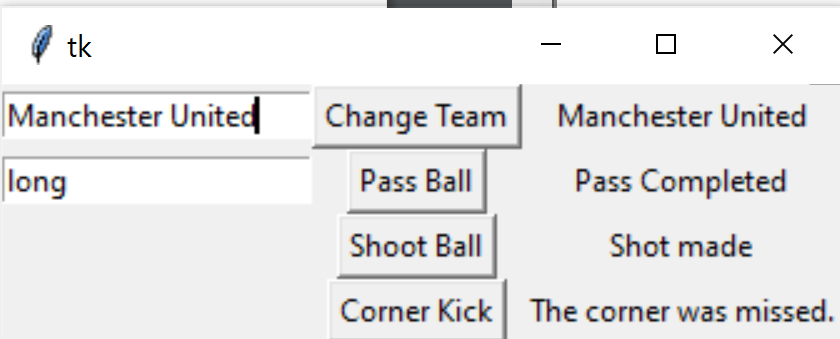
1. After passing, the user can choose to take a shot. The probability for making a shot is based on how many passes have been made, and how hard the passes were to make.
   1. For example, completing three ‘long’ passes will result in a better probability that a shot will score compared to three ‘short’ passes.
2. The user can then choose to take a corner kick. If a shot has been taken and missed, the corner kick will be allowed. However, if the shot was made, the user will not be able to take a corner.
   1. If the user makes a shot before attempting to take a corner, the program will tell the user that a shot has already been made.

**Screenshots with Sample Input/Output:**

****

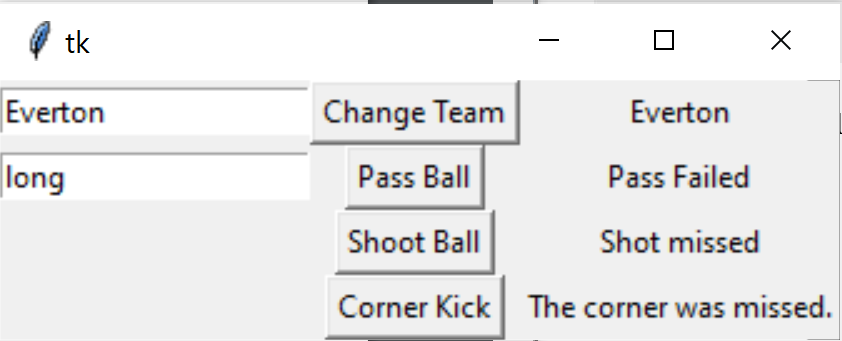
Input: Manchester City, ‘long’ pass

Output: Failed pass, missed shot, missed corner

****

Input: Manchester United, ‘long’ pass

Output: Completed pass, made shot

****

Input: Everton, ‘long’ pass

Output: Failed pass, missed shot, missed corner

**APPENDIX**

**MAIN:**

|  |
| --- |
| **# -\*- coding: utf-8 -\*- ''' =============================================================================== ENGR 133 Program Description   creates widget, runs other defined functions to create word based soccer game  Assignment Information  Assignment: Individual Project Main   Author: Heath Aaron Lovell, hlovell@purdue.edu  Team ID: 003-15 (e.g. 001-14 for section 1 team 14)   Contributor: N/A  My contributor(s) helped me:   [ ] understand the assignment expectations without  telling me how they will approach it.  [ ] understand different ways to think about a solution  without helping me plan my solution.  [ ] think through the meaning of a specific error or  bug present in my code without looking at my code.  Note that if you helped somebody else with their code, you  have to list that person as a contributor here as well. =============================================================================== ''' #Importing all the needed modules and user defined functions import tkinter as tk import numpy as np import PyIndivProj\_passBall as passingBall import PyIndivProj\_shootBall as shootingBall import PyIndivProj\_cornerKick as cornerKick  root = tk.Tk()  #Initialize the team name entry variable teamNameEntryVar = tk.StringVar() teamNameEntryVar.set("Enter EPL team name")  #Initialize the pass distance variable distanceVar = tk.StringVar() distanceVar.set('"short" or "long" pass')  #Initialize the team name variable teamName = tk.StringVar() teamName.set("Your team")  #Initialize the pass complete variable passComplete = tk.StringVar() passComplete.set("Pass Results")  #Initialize the pass values vector passValVector = []  #Initialize the complete shot variable completeShot = tk.StringVar() completeShot.set("Shot Results")  #Initialize the corner result variable cornerResult = tk.StringVar() cornerResult.set("Corner Kick Results")  #Initialize the list of teams the user can choose teamNames = ["Arsenal", "Aston Villa", "AFC Bournemouth", "Brighton and Hove Albion", "Burnley", "Chelsea", "Crystal Palace", "Everton", "Leicester City", "Liverpool", "Manchester City", "Manchester United", "Newcastle United", "Norwich City", "Sheffield United", "Southampton", "Tottenham Hotspur", "Watford", "West Ham United", "Wolverhampton Wanderers"]  #Define the function allowing the user to set the team def setTeam():  global teamName  teamNameInput = teamNameEntry.get() #Gets the user input in the team name entry  if teamNameInput in teamNames: #Checks to make sure that the input is in the valid list of team names  teamName.set(teamNameInput) #Sets the team name label to the input from user  else:  while teamNameInput not in teamNames: #Runs loop for when user input is not in the valid list of team names  teamNameFix = tk.simpledialog.askstring('Invalid Name', 'Please enter a valid team name.') #Gets new user input, asking for valid team name    for i in range(len(teamNames)): #Runs loop for every value in the valid team names list  if teamNameFix == teamNames[i]: #Checks to see if the team name is in the valid list of names  teamName.set(teamNameFix) #If the input is valid, it sets the team name variable to the input  teamNameEntryVar.set(teamNameFix) #If input is valid, sets team name entry to input  teamNameInput = teamNameFix #Sets team name input to the valid name   #Defines the pass ball function  def passBall():  distance = passDistanceEntry.get() #Gets user input for 'short' or 'long' pass  if distance == "short" or distance == "long": #Making sure user input is valid   passOutput = passingBall.passComplete(distance) #Sets pass output the the value returned by the function 'passComplete'  passComplete.set(passOutput[0]) #Sets the pass complete label variable to the pass output  passValVector.append(passOutput[1]) #Sets the pass value vector to the vector returned by the function 'passComplete'  elif distance != "short" and distance != "long": #If distance is not valid  while distance != "short" and distance != "long": #Loop for when distance is not valid  newDistance = tk.simpledialog.askstring('Invalid distance.', 'Please enter either "short" or "long".') #Prompts user to enter a new valid distance  distance = newDistance #Sets distance to the valid distance    distanceVar.set(distance) #Sets distance entry to distance inputted by user   #Defines the shoot ball function def shootBall():   global shotValue  shotValue = shootingBall.shotComplete(passValVector) #Sets shot value to the value returned by the 'shotComplete' function  completeShot.set(shotValue) #Sets the complete shot label to the shot value   #Defines the corner kick function def cornerKickBall():  resultCorner = cornerKick.cornerComplete(shotValue) #Sets the corner result to the value returned by the 'cornerComplete' function  cornerResult.set(resultCorner) #Sets the corner result label to the corner result   #Defines 0 of the widget teamNameEntry = tk.Entry(root, text = teamNameEntryVar, textvariable = teamNameEntryVar) changeTeamBtn = tk.Button(root, text='Change Team', command=setTeam) teamNameLbl = tk.Label(root, text = teamName, textvariable = teamName)  #Defines 1 of the widget passDistanceEntry = tk.Entry(root, text = distanceVar, textvariable = distanceVar) passBallBtn = tk.Button(root, text='Pass Ball', command=passBall) passCompleteLbl = tk.Label(root, text = passComplete, textvariable = passComplete)  #Defines 2 of the widget shootBallBtn = tk.Button(root, text = 'Shoot Ball', command=shootBall) shootBallLbl = tk.Label(root, text = completeShot, textvariable = completeShot)  #Defines 3 of the widget cornerKickBtn = tk.Button(root, text = "Corner Kick", command=cornerKickBall) cornerKickLbl = tk.Label(root, text = cornerResult, textvariable = cornerResult)  #Organizes row 0 in the widget teamNameEntry.grid(row=0, column=0) changeTeamBtn.grid(row=0, column=1) teamNameLbl.grid(row=0, column=2)  #Organizes row 1 in the widget passDistanceEntry.grid(row=1, column=0) passBallBtn.grid(row=1, column=1) passCompleteLbl.grid(row=1, column=2)  #Organizes row 2 in the widget shootBallBtn.grid(row=2, column=1) shootBallLbl.grid(row=2, column=2)  #Organizes row 3 in the widget cornerKickBtn.grid(row=3, column=1) cornerKickLbl.grid(row=3, column=2)  root.mainloop()  ''' =============================================================================== ACADEMIC INTEGRITY STATEMENT  I have not used source code obtained from any other unauthorized  source, either modified or unmodified. Neither have I provided  access to my code to another. The project I am submitting  is my own original work. =============================================================================== '''** |

**UDF – Pass:**

|  |
| --- |
| **# -\*- coding: utf-8 -\*- ''' =============================================================================== ENGR 133 Program Description   defines function to determine whether pass is completed or failed  Assignment Information  Assignment: Individual Project Pass Ball  Author: Heath Aaron Lovell, hlovell@purdue.edu  Team ID: 003-15 (e.g. 001-14 for section 1 team 14)   Contributor: N/A  My contributor(s) helped me:   [ ] understand the assignment expectations without  telling me how they will approach it.  [ ] understand different ways to think about a solution  without helping me plan my solution.  [ ] think through the meaning of a specific error or  bug present in my code without looking at my code.  Note that if you helped somebody else with their code, you  have to list that person as a contributor here as well. =============================================================================== ''' #Imports needed modules import numpy as np  #Defines function to determine whether pass was completed or not def passComplete(distance):  randPass = np.random.randint(1,10) #Sets randPass to a random value    if distance == "short": #Checks if the distance entered by user is 'short'  if randPass > 2: #Checks if the randPass value is greater than 2  completePass = "Pass Completed" #Sets variable to tell main program that the pass was complete  passValue = 10 #The value of a short pass is 10  return completePass, passValue #Returns two variables to the main program  else:  failedPass = "Pass Failed" #Sets variable to tell main program that the pass failed  passValue = 0 #The value of a failed pass is 0  return failedPass, passValue #Returns two variables to the main program  elif distance == "long": #Checks if the distance entered by user is 'long'  if randPass > 7: #Checks if the randPass value is greater than 7  completePass = "Pass Completed" #Sets variable to tell main program that the pass was complete  passValue = 30 #The value of a long pass is 30  return completePass, passValue #Returns two variables to the main program  else:  failedPass = "Pass Failed" #Sets variable to tell main program that the pass failed  passValue = 0 #The value of a failed pass is 0  return failedPass, passValue #Returns two variables to the main program   ''' =============================================================================== ACADEMIC INTEGRITY STATEMENT  I have not used source code obtained from any other unauthorized  source, either modified or unmodified. Neither have I provided  access to my code to another. The project I am submitting  is my own original work. =============================================================================== '''** |

**UDF – Shoot:**

|  |
| --- |
| **# -\*- coding: utf-8 -\*- ''' =============================================================================== ENGR 133 Program Description   defines function to determine whether shot is missed or made  Assignment Information  Assignment: Individual Project Shoot Ball  Author: Heath Aaron Lovell, hlovell@purdue.edu  Team ID: 003-15 (e.g. 001-14 for section 1 team 14)   Contributor: N/A  My contributor(s) helped me:   [ ] understand the assignment expectations without  telling me how they will approach it.  [ ] understand different ways to think about a solution  without helping me plan my solution.  [ ] think through the meaning of a specific error or  bug present in my code without looking at my code.  Note that if you helped somebody else with their code, you  have to list that person as a contributor here as well. =============================================================================== ''' #Imports needed modules import numpy as np  #Defines function to determine whether or not the shot is made def shotComplete(passValVector):  randShot = np.random.randint(1,100) #Sets randShot to a random value  shotPercentage = (.3 \* sum(passValVector)) #The percentage that the shot will be made is set to a value, increasing with the amount of passes complete  if randShot < shotPercentage: #Checks to see if the random value is less than the shot percentage  shotResult = "Shot made" #Sets the variable to tell the main program that the shot was made  else: #Runs if the random value is not less than the shot percentage  shotResult = "Shot missed" #Sets the variable to tell the main program that the shot was missed  return shotResult #Returns the variable telling the result of the shot  ''' =============================================================================== ACADEMIC INTEGRITY STATEMENT  I have not used source code obtained from any other unauthorized  source, either modified or unmodified. Neither have I provided  access to my code to another. The project I am submitting  is my own original work. =============================================================================== '''** |

**UDF – Corner:**

|  |
| --- |
| **# -\*- coding: utf-8 -\*- ''' =============================================================================== ENGR 133 Program Description   defines function to determine whether corner kick is missed or made  Assignment Information  Assignment: Individual Project Corner Kick Function  Author: Heath Aaron Lovell, hlovell@purdue.edu  Team ID: 003-15 (e.g. 001-14 for section 1 team 14)   Contributor: N/A  My contributor(s) helped me:   [ ] understand the assignment expectations without  telling me how they will approach it.  [ ] understand different ways to think about a solution  without helping me plan my solution.  [ ] think through the meaning of a specific error or  bug present in my code without looking at my code.  Note that if you helped somebody else with their code, you  have to list that person as a contributor here as well. =============================================================================== ''' #Imports needed modules import numpy as np  #Defines function to determine whether a corner kick is made def cornerComplete(shotValue):  if shotValue == "Shot made": #Checks to see if the previous shot attempt was made  resultCorner = "The shot was made" #Sets the return variable to tell the user that the shot was made  else: #Runs if the shot was not made  randCorner = np.random.randint(1,10) #Sets randCorner variable to a random number  if (randCorner < 6): #If the random number is less than 6, the corner is made  resultCorner = "The corner was made!"  else:  resultCorner = "The corner was missed." #If the random number is not less than 6, the corner was missed  return resultCorner #Returns the variable that tells the main program whether or not the corner was made  ''' =============================================================================== ACADEMIC INTEGRITY STATEMENT  I have not used source code obtained from any other unauthorized  source, either modified or unmodified. Neither have I provided  access to my code to another. The project I am submitting  is my own original work. =============================================================================== '''** |