

12/6/2011



FANTASY BASKETBALL SIMULATOR

Colin Robinson

Table of Contents

Executive Summary	iii
Introduction	1
Instructions: Simulating	1
Instructions: Adding New Players	4
Practical Uses	5
Implementation	6
Complications	10
Learning	11
Quick Reference Guide	12

Executive Summary

Purpose

The Fantasy Basketball Simulator was created to help owners of fantasy basketball teams to optimize their teams and lineups in order to give them a competitive advantage over other teams in their leagues and save them countless hours of analysis.

Overview

Fantasy Basketball is a competition where participants own, manage, and coach imaginary teams and compete with other participants to achieve the highest scores based on statistics generated by actual players of the National Basketball Association. This specific program was designed to be used in Head-to-Head leagues, where each week teams are scheduled to compete against another team in the league based on seven statistics: points, field goal percentage, rebounds, assists, blocks, turnovers, and steals. Teams consist of 12 players but are limited to only 7 players which are allowed to compete in any given week. A team wins the weekly matchup when its players outperform its opponent's players on at least 4 of the 7 previously mentioned statistics during the week.

There are three main steps for owners as they seek to effectively use the program:

1. **Input Data** – The user must input the names of the players on their roster into column C and their opponent's roster into column I as well as the number of games each player has during the week being analyzed into columns E and K.
2. **Select Lineups** – The user clicks the **Simulate** button after which a message box will come up prompting the user to select the seven players on each team which will compete for the specific week.
3. **Analyze Results** – The results of the simulation will be printed in the **Simulation Results** box. The user can then analyze these results and make adjustments to the lineup to optimize his or her team.

Practical Uses

The Fantasy Basketball Simulator is intended to be used as an aid to fantasy basketball team owners in decision making for their teams. There are three main decisions the program can assist with:

1. **Optimizing Lineups** – Owners can predict their opponent's team and then simulate several variations of their own team in order to find the combination of players that has the highest probability of winning for that specific week.
2. **Analyzing Trades and Free Agents Pickups** – Owners can replace players from their current roster with players that they can potentially acquire via trades or free agency to find the combination of players that has the highest probability of winning.
3. **Drafting Players** – Owners can simulate teams with players they intend to target in the league's draft to determine value added to the team and optimal team combinations.

Fantasy Basketball Simulator Write-Up

Introduction

For every individual that is athletic and talented enough to make a living playing sports, there are at least a thousand others who only wish they were. For those dreamers, there exists a special sport that allows participants to fantasize for a moment that they are in the shoes of their favorite athletes, coaches, and team managers. That special sport is called fantasy basketball. The popularity of fantasy basketball has been increasing in various social settings, such as among family members, friends, and coworkers; because of this rise in popularity, it is becoming increasingly important that participants can competently manage a fantasy team to gain credibility and bragging rights over their peers.

Fantasy Basketball is a competition where participants own, manage, and coach imaginary teams and compete with other participants to achieve the highest scores based on statistics generated by actual players of the National Basketball Association. Leagues generally consist of 8 to 16 fantasy teams with 12 actual NBA players on each team. Team rosters are selected by participants at the beginning of the competition in an event known as the draft. In the following weeks, participants set lineups, make trades, and add free agents to optimize their lineups as they compete to have the best statistics.

This Fantasy Basketball Simulator was designed specifically to be used in “Head-to-Head” leagues, where each week teams are scheduled to compete against another specific team in the league. Owners may choose 7 of their 12 players to play each week in the matchup and must choose at least one player that is capable of playing each of the five basic positions: point guard, shooting guard, small forward, power forward, and center. The two teams compete to have better statistics across seven categories: points, field goal percentage, rebounds, assists, blocks, turnovers, and steals. A team wins the weekly matchup when its players outperform its opponent’s players on at least 4 of the 7 statistical categories during the week.

The Fantasy Basketball Simulator allows the user the pick the seven players that will compete in the weekly matchup, and then will retrieve updated season averages of each player from the internet to calculate the probable results of the hypothetical matchup. Often owners spend hours each week manually retrieving data and making calculations to optimize their teams. This program will be able to retrieve more accurate results for them and save them countless hours of analysis throughout the basketball season.

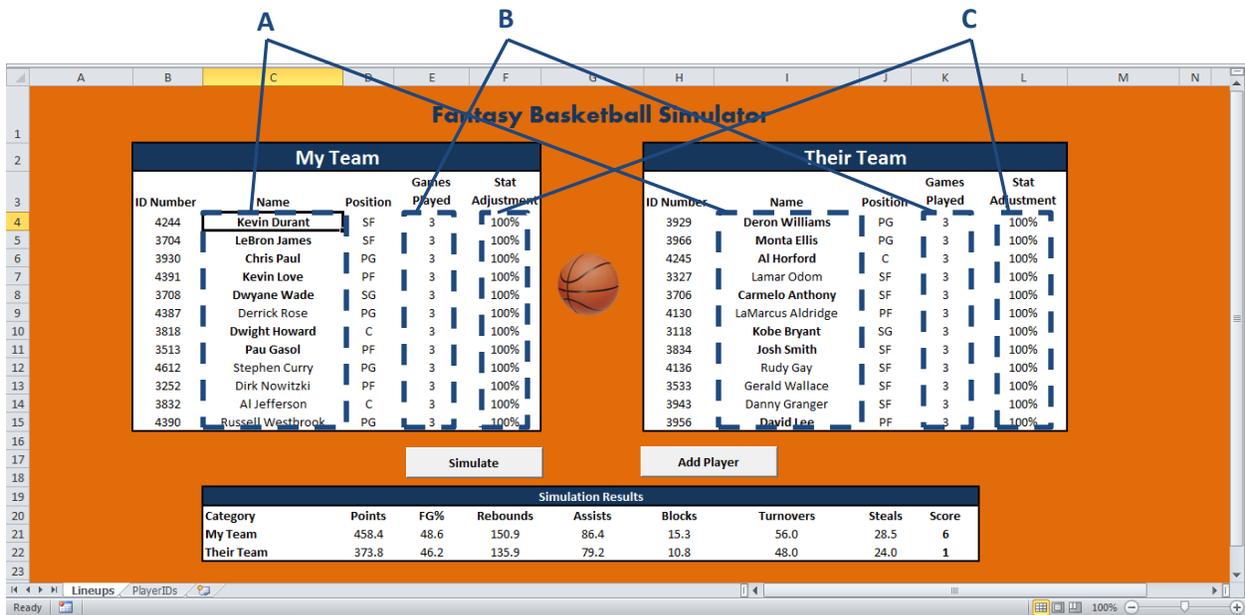
Instructions: Simulating

(See also: Quick Reference Guide)

The Fantasy Basketball Simulator was designed to be as user-friendly and intuitive as possible. There are three main steps for owners as they seek the effectively use the program:

1. **Input Data** (See diagram below)

- a. **Player Names** – The user starts by inputting the names of the players on their roster into column C and their opponent’s roster into column I. The user must type the names of the players correctly because the program uses a vlookup function to determine the players’ identification numbers and positions.
- b. **Games Played** – The user then enters the number of games each player has during the week being analyzed into columns E and K. This number is used to calculate the player’s weekly performance. For example, a player who plays four games in a week will likely produce twice as many stats as a player with only two games in the week.
- c. **Stat Adjustment** – The user then has the option of adjusting the percentages used to calculate the expected statistics of each player in columns F and L. For example, if the user has a player that is returning from an injury and likely will not play as many minutes as he normally does then the user may choose to lower the stat adjustment percentage to below 100%. This will cause the simulator to take into account the decreased performance of the player by multiplying the player’s season averages by the stat adjustment percentage. The stat adjustment percentage can also be increased to over 100% to potentially take into account a player that has been over performing recently.

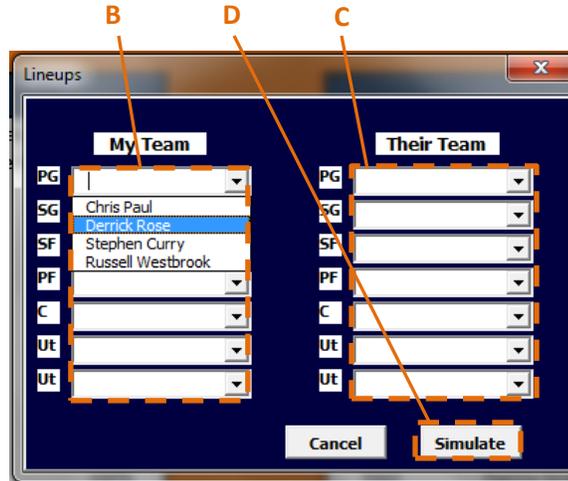


2. Select Lineups (See diagram below)

- a. **Click the Simulate button** – A message box will appear allowing the user to select the lineups for the simulation.
- b. **My Team Lineup** – The user next chooses 7 players from his or her team of 12 to simulate the week’s matchup. The user must choose one player for each of the five standard positions as well as two “utility” players that can play any position. Note that

the user may not choose utility players that have already been picked as starting five players.

- c. **Their Team Lineup** – The user then follows the same process to choose 7 players for the opposing team. For best results, the user should attempt to predict the lineup that his or her opponent will most likely use.
- d. **Click the Simulate Button** – The user finishes by clicking the simulate button on the bottom-right corner of the message box.



- 3. **Analyze Results** – The program will then use the internet to look up recent season averages for each of the players chosen (note that this step may take 1-2 minutes). It will then calculate the sum of the season averages of each player multiplied by the games played in the week and the stat adjustment percentage. The results of the calculation will be printed in the "Simulation Results" box (See diagram below). The box displays the expected total statistics of both teams in each category as well as a final score out of seven for each team depending on which categories they outperformed their opponents in. The user can then analyze these results and repeat the process making adjustments to the lineup to optimize team performance.

Fantasy Basketball Simulator

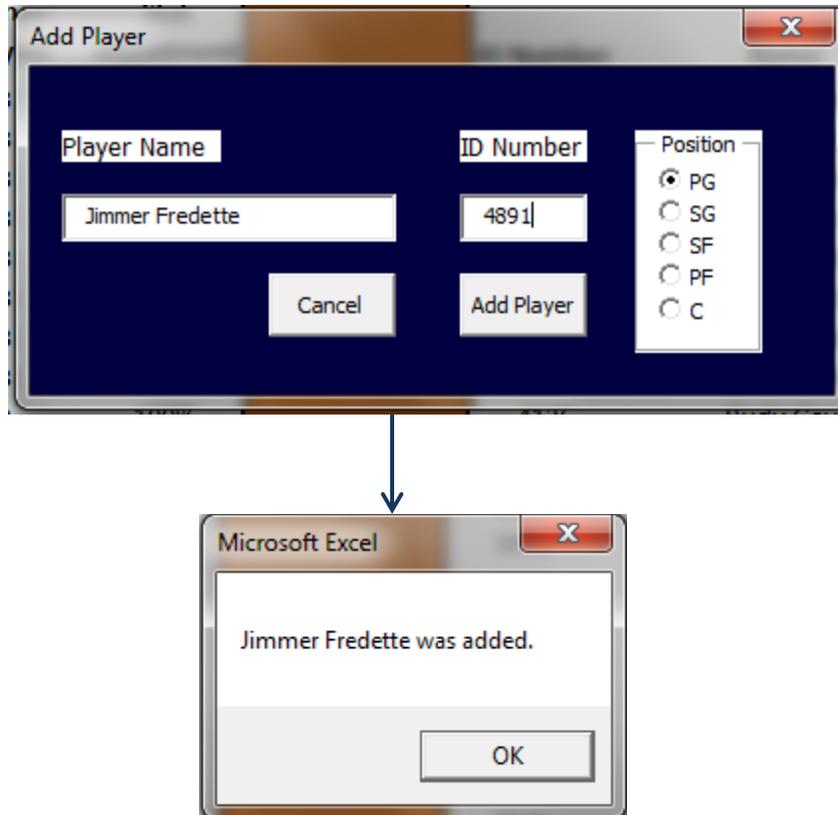
My Team					Their Team				
ID Number	Name	Position	Games Played	Stat Adjustment	ID Number	Name	Position	Games Played	Stat Adjustment
4244	Kevin Durant	SF	3	100%	3929	Deron Williams	PG	3	100%
3704	LeBron James	SF	3	100%	3966	Monta Ellis	PG	3	100%
3930	Chris Paul	PG	3	100%	4245	Al Horford	C	3	100%
4391	Kevin Love	PF	3	100%	3327	Lamar Odom	SF	3	100%
3708	Dwyane Wade	SG	3	100%	3706	Carmelo Anthony	SF	3	100%
4387	Derrick Rose	PG	3	100%	4130	LaMarcus Aldridge	PF	3	100%
3818	Dwight Howard	C	3	100%	3118	Kobe Bryant	SG	3	100%
3513	Pau Gasol	PF	3	100%	3834	Josh Smith	SF	3	100%
4612	Stephen Curry	PG	3	100%	4136	Rudy Gay	SF	3	100%
3252	Dirk Nowitzki	PF	3	100%	3533	Gerald Wallace	SF	3	100%
3832	Al Jefferson	C	3	100%	3943	Danny Granger	SF	3	100%
4390	Russell Westbrook	PG	3	100%	4396	David Lee	PF	3	100%

Simulation Results								
Category	Points	FG%	Rebounds	Assists	Blocks	Turnovers	Steals	Score
My Team	458.4	48.6	150.9	86.4	15.3	56.0	28.5	6
Their Team	373.8	46.2	135.9	79.2	10.8	48.0	24.0	1

Instructions: Adding New Players

The program uses a table located on the "PlayerIDs" tab to know where to look on the internet to retrieve updated statistics for each player and to find out what positions they are capable of playing. The table currently contains 150 of the most frequently used fantasy basketball players with their identification numbers and positions. If the user desires to simulate with a player that is not currently located on the table, he or she will first need to add the player to the table (note that if the user attempts to simulate with an unknown player the program will respond with a message informing the user that the player must be added to the PlayerIDs tab). The following outlines the steps necessary to add a player to the table (see diagram below):

1. Click the "Add Player" button
2. Type the name of the player under "Player Name"
3. Retrieve the identification number of the player
 - a. Go to <http://sports.yahoo.com/nba/players> and perform a search for the player
 - b. Click on the player's link and note the last four digits of the new URL
4. Type the player's identification number under "ID"
5. Click the option circle corresponding to the player's position
6. Click the "Add Player" button when finished
7. A message should appear indicating to the user that the player was successfully added
8. The user may not run the simulation with that player



*Note that rookies like Jimmer cannot currently run in the program because they have no NBA statistics

Practical Uses

The Fantasy Basketball Simulator is intended to be used as an aid to fantasy basketball team owners in decision making for their teams. There are three main decisions the program can assist with:

1. **Optimizing Lineups** – Owners can predict their opponent’s team and then simulate several variations of their own team to find the combination of players that has the highest probability of winning for that specific week.
2. **Analyzing Trades and Free Agents Pickups** – Owners can replace players from their current roster with players that they can potentially acquire via trades or free agency to find the combination of players that has the highest probability of winning.
3. **Drafting Players** – Owners can simulate teams with players they intend to target in the league’s draft to determine value added to the team and optimal team combinations.

Using the program certainly does not guarantee that the user will win his or her fantasy league, but it does guarantee that he or she will make better decisions and use time more efficiently.

Implementation

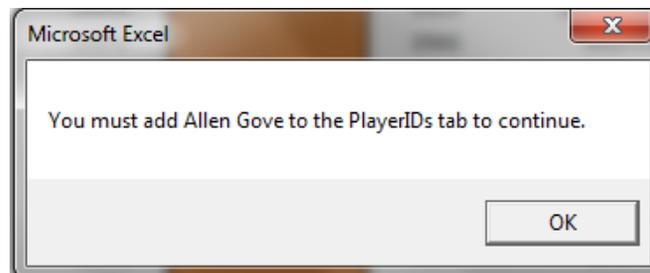
This section gives a more technical perspective of the program and how it functions.

Inputting the Data. The first major step to running the program is ensuring that all of the data listed on the “My Team” and “Their Team” sections are accurately completed. As mentioned in the instructions section, it is the user’s responsibility to fill out columns C, E, F, I, K, and L with the players’ names, games played in the week, and stat adjustment percentages. Columns B, D, H, and J will update automatically with the players’ identification numbers and positions. This data is retrieved from the “PlayerIDs” tab using a VLOOKUP function. If the user inputs a player’s name that is not listed in the “PlayerIDs” table then the user must first click the “Add Player” button and enter the players name, identification number, and position before simulating (See Adding New Players section). After the data is successfully entered, the worksheet should look like the following diagram.

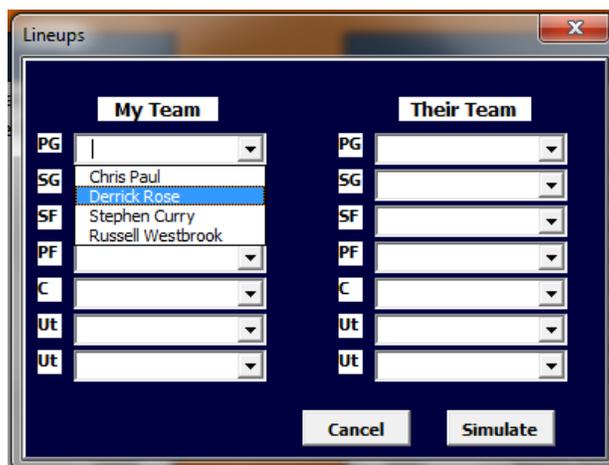
My Team					Their Team				
ID Number	Name	Position	Games Played	Stat Adjustment	ID Number	Name	Position	Games Played	Stat Adjustment
4244	Kevin Durant	SF	3	100%	3929	Deron Williams	PG	3	100%
3704	LeBron James	SF	3	100%	3966	Monta Ellis	PG	3	100%
3930	Chris Paul	PG	3	100%	4245	Al Horford	C	3	100%
4391	Kevin Love	PF	3	100%	3327	Lamar Odom	SF	3	100%
3708	Dwyane Wade	SG	3	100%	3706	Carmelo Anthony	SF	3	100%
4387	Derrick Rose	PG	3	100%	4130	LaMarcus Aldridge	PF	3	100%
3818	Dwight Howard	C	3	100%	3118	Kobe Bryant	SG	3	100%
3513	Pau Gasol	PF	3	100%	3834	Josh Smith	SF	3	100%
4612	Stephen Curry	PG	3	100%	4136	Rudy Gay	SF	3	100%
3252	Dirk Nowitzki	PF	3	100%	3533	Gerald Wallace	SF	3	100%
3832	Al Jefferson	C	3	100%	3943	Danny Granger	SF	3	100%
4390	Russell Westbrook	PG	3	100%	3956	David Lee	PF	3	100%

Simulation Results								
Category	Points	FG%	Rebounds	Assists	Blocks	Turnovers	Steals	Score
My Team								
Their Team								

Loading the User Form. Next the user clicks the simulate button which will open the user form called “Simulate.” If one of the players listed in columns C or I have not been entered into the “PlayerIDs” table then a message box will appear informing the user to add the player before continuing. The program checks by making sure that the numbers listed in columns B and H are greater than zero. If not the corresponding name is entered into a variable called “add player” which is used to produce the message box shown to the user (see example below).

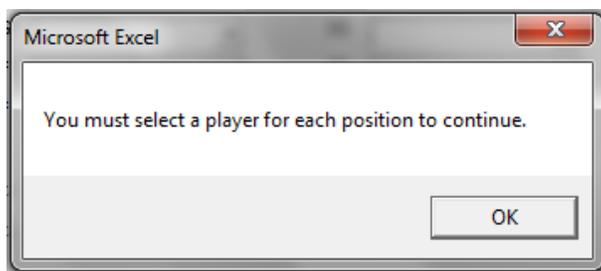


Using the User Form. When all the names and data are accurately inputted, the “Simulate” user form will appear to the user. The box includes two columns of seven ComboBoxes where the user may select the player they wish to use for each position (see diagram below).

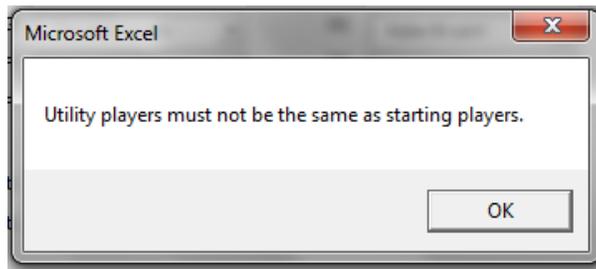


The ComboBoxes for each position are populated only with the names of the players that are capable of playing that position. The program uses a loop for each box and loops through cells D4:D15 for the “My Team” boxes and J4:J15 for the “Their Team” boxes. When the program finds a player that matches the position, the name in the cell to the left is then added as an option to the ComboBox. The “Utility” positions are populated with all of the players on the corresponding teams. When the player is finished selecting the players to simulate in the matchup, he or she then clicks on the simulate button.

If the user does not select a player for every position then the following message will be displayed (see diagram below). The program goes through and checks that the value selected for each ComboBox is not empty to determine if the user has selected a player for each position.



If the user selects the same player for a utility and starting position then the following message will be displayed (see diagram below). The program uses an “if statement” to make sure that all of the values selected in the utility ComboBoxes are not the same values as the ones selected in the position ComboBoxes.



The user may also click the cancel button or the “x” button to make the user form disappear.

Identifying the Selected Players. After the players are correctly chosen and the “Simulate” button is clicked, the program then looks to identify the players that were chosen by the user for the simulation. The program takes the value selected in the ComboBox and searches for it on the worksheet. When it identifies the location of the player’s name on the worksheet, it then saves the row number as a variable. The program uses a loop to repeat the process for each of the 14 positions saving the row numbers with names indicative of the positions they represent such as “PGNum”, “C2Num” or “UTa2Num”. The program also bolds the names of the 14 players selected as it finds them which is helpful to the user as he or she analyzes the results.

Retrieving the Data. The program next uses the internet to find the current season statistics for each of the players on both teams. The program starts by adding a new worksheet to the workbook. Screen updating is turned off so that the user is unaware of this. Next the program uses internet explorer to bring up the following URL: “<http://sports.yahoo.com/nba/players/>” & ID. The program uses a variable called ID to bring up the page specific to the player it is looking for.

It then prints the data of a specific table from the webpage to the new worksheet. The code then checks to make sure there are numbers populated in the correct cells to ensure that the right data was retrieved from the web page. Using “If statements”, the program continues checking and printing tables until the correct data is uploaded.

The program then uploads the values of seven of the cells in the new worksheet to seven different arrays corresponding to each statistic that is measured in the simulation. After the data is uploaded in the arrays, the worksheet is deleted. This process is then looped through 24 times to retrieve and save updated statistics from the internet for all of the players. This step can take between one to three minutes to execute completely.

Calculating the Results. The program next uses the retrieved data and the data from the user to calculate the scores for each of the seven statistical categories. These lines of code use simple mathematical formulas using variables to calculate each of the teams’ scores for each of the seven categories. The following line is an example of the code used to calculate “My Team’s” points:

$$\text{MyPoints} = \text{Points}(\text{PGNum}) * \text{Cells}(\text{PGNum} + 4, 5).\text{Value} * \text{Cells}(\text{PGNum} + 4, 6).\text{Value} +$$

$$\text{Points}(\text{SGNum}) * \text{Cells}(\text{SGNum} + 4, 5).\text{Value} * \text{Cells}(\text{SGNum} + 4, 6).\text{Value} + \text{Points}(\text{SFNum}) *$$

$$\text{Cells}(\text{SFNum} + 4, 5).\text{Value} * \text{Cells}(\text{SFNum} + 4, 6).\text{Value} + \text{Points}(\text{PFNum}) * \text{Cells}(\text{PFNum} + 4,$$

$$5).\text{Value} * \text{Cells}(\text{PFNum} + 4, 6).\text{Value} + \text{Points}(\text{CNum}) * \text{Cells}(\text{CNum} + 4, 5).\text{Value} *$$

$$\text{Cells}(\text{CNum} + 4, 6).\text{Value} + \text{Points}(\text{UtaNum}) * \text{Cells}(\text{UtaNum} + 4, 5).\text{Value} * \text{Cells}(\text{UtaNum} +$$

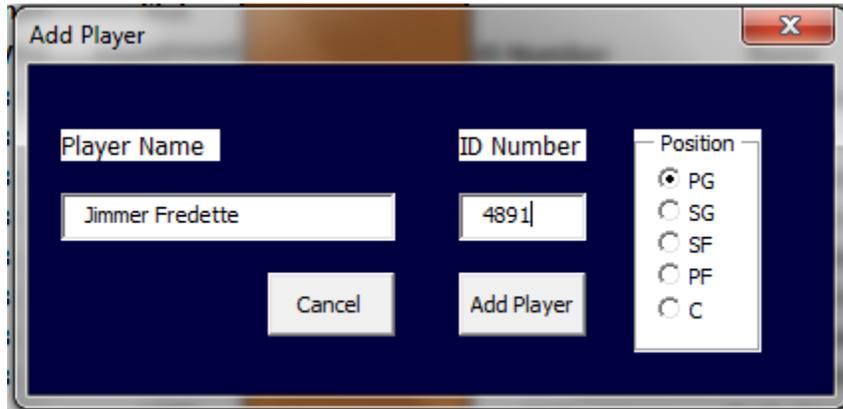
$$4, 6).\text{Value} + \text{Points}(\text{UtbNum}) * \text{Cells}(\text{UtbNum} + 4, 5).\text{Value} * \text{Cells}(\text{UtbNum} + 4, 6).\text{Value}$$

As can be seen in the code, the calculation finds the sum of the array value for the each player multiplied by the number of games played and the stat adjustment percentage to calculate the score. The program runs a similar code for each category for both teams and uploads the calculations into variables. Then the program uses “If” statements to calculate the total matchup score. For example, the code “If MyPoints > TheirPoints Then MyScore = MyScore + 1” increases the user’s team’s score if his or her team had a higher point total than the opposing team. The opponent’s score is calculated by subtracting “MyScore” from seven.

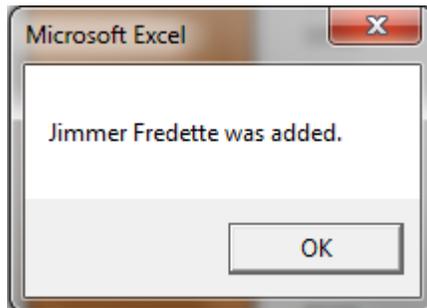
Outputting the Results. Lastly the program prints the scores of the seven statistics and the total score for each team in cells D21:K22. As mentioned above, the names of the players selected by the user will appear in bold to remind the user which players were chosen when analyzing the results. The finished spreadsheet should appear as follows (see diagram below).



Adding Players. As mentioned previously, the simulation will only run correctly when all of the players are listed in the “PlayerIDs” tab with their identification numbers and positions. The user may simply add a new player to the table by clicking on the “Add Player” button. This button will open a new user form called “Add Player” which includes two text boxes and an option button. Here the user can type in the name and identification number of the player and click on the option button corresponding to the player’s position (see diagram below).



When finished, the user clicks the “Add Player” button. The program then opens the “PlayerIDs” tab and uses ctrl + shift + down to reach the bottom of the table. The program then enters the values of the of the text boxes into the cells in column A and B; the program uses “If statements” to determine the position chosen for the player and prints the position into column C of the table. The program finishes by returning to the “Lineups” tab and informing the user through a message box that the player was added.



Complications

I came across a couple of issues that were particularly difficult to solve when creating this program.

Internet Formatting. The program uses Internet Explorer to go online and retrieve updated season averages for each player in the simulation. In most cases, the relevant information for the player was located in the same position for each player. However, there were a few exceptions. I noticed some players were causing errors in the program or were causing the program to miscalculate the statistics. After further investigation, I noticed that some of the players have web pages that are formatted slightly different than the rest.

To solve this problem, I had to write a loop within the loop that retrieved data that allowed the program to retrieve and test several table locations within the web page until it had retrieved the correct table. Luckily, the table that the program needed to upload was larger than the other tables on the web page

so I could loop the process until a particular cell within the worksheet I was printing to had a value greater than zero.

Program Testing. The second issue I came across while making this program was during testing. Because it took so long to retrieve the data for all the players from the internet (sometimes longer than three minutes with a slow internet connection), it made testing my code very time consuming and difficult. I was able to designate this long process to its own sub procedure and turn it off while I tested other parts of the code, but many times it was necessary to run the data retrieval loop to make sure the whole program was running as expected. There was no complete solution to this problem, except for patience and a lot of time.

Learning

I learned many things about VBA programming while completing this project.

One of the most important skills I developed was being able to recognize a need and then conceptualize a solution to the need through VBA. This is a skill I was not able to develop in class on previous projects because they outlined exactly what we needed to do with little room for added creativity. This is the first time I had to think about what I wanted the project to accomplish and make a plan to achieve my desired results.

I think that this skill of recognizing the problem and conceptualizing the solution is very important in business. In my career, I am not expecting managers to outline exactly what the problem is and how it is to be solved. I believe that the skills I developed from this project will help me find innovative solutions to problems in my future career.

I also learned and developed many technical programming skills by working on this project. I am now much better at working with arrays, using VBA to retrieve data from the internet, using loops and if statements, and creating user forms. Overall, I am very glad that I chose this as my project, and I am very pleased with my results from a results perspective and a learning perspective.

Quick Reference Guide

Player Names (C4:C15, I4:I15)

Here the user enters the names of his or her players and the names of the players on the opposing team

Games Played (E4:E15, K4:K15)

Here the user enters the number of games played in the week by each player listed in cells C4:C15 and I4:I15

Stat Adjustment (F4:F15, L4:L15)

Here the user may optionally enter a percentage of the player's season average statistics to be used in the calculation of the simulation

My Team					Their Team				
ID Number	Name	Position	Games Played	Stat Adjustment	ID Number	Name	Position	Games Played	Stat Adjustment
4244	Kevin Durant	SF	3	100%	3929	DeRon Williams	PG	3	100%
3704	LeBron James	SF	3	100%	3966	Monta Ellis	PG	3	100%
3930	Chris Paul	PG	3	100%	4245	Al Horford	C	3	100%
4391	Kevin Love	PF	3	100%	3327	Lamar Odom	SF	3	100%
3708	Dwyane Wade	SG	3	100%	3706	Carmelo Anthony	SF	3	100%
4387	Derrick Rose	PG	3	100%	4130	LaMarcus Aldridge	PF	3	100%
3818	Dwight Howard	C	3	100%	3118	Kobe Bryant	SG	3	100%
3513	Pau Gasol	PF	3	100%	3834	Josh Smith	SF	3	100%
4612	Stephen Curry	PG	3	100%	4136	Rudy Gay	SF	3	100%
3252	Dirk Nowitzki	PF	3	100%	3533	Gerald Wallace	SF	3	100%
3832	Al Jefferson	PG	3	100%	3943	Danny Granger	SF	3	100%
4390	Russell Westbrook	C	3	100%	3956	David Lee	PF	3	100%

Simulation Results								
Category	Points	EG%	Rebounds	Assists	Blocks	Turnovers	Steals	Score
My Team	456.4	48.6	150.9	86.4	15.3	56.0	28.5	6
Their Team	373.8	46.2	135.9	79.2	10.8	48.0	24.0	1

Simulate Buttons

The user presses this button to run the simulation program after lineups have been properly set

Position Drop-Boxes

Here the user selects the seven players on each team that will be used for the simulation

Simulation Results (C21:K22)

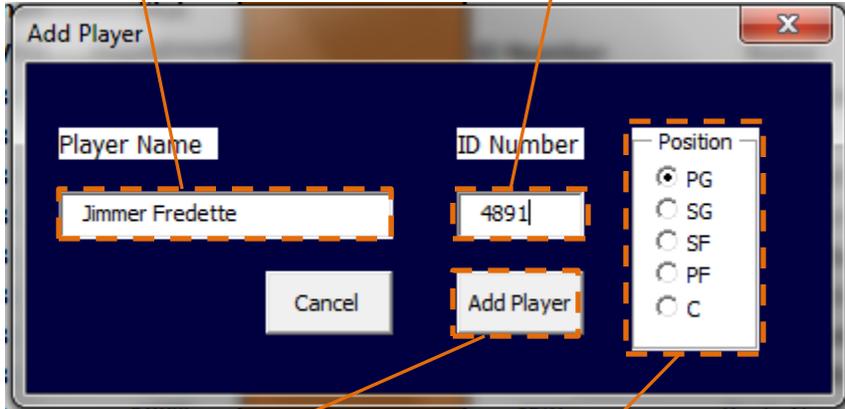
The results of the simulation will be printed here for the user to analyze

Add Player Button

The user clicks this button to add a new player to the simulator database

Player Name Text Box
Here the user inputs the name of the player to be used for the simulation

Player ID Number Text Box
Here the user inputs the ID number of the player to be used for the simulation



Add Player Button
The user clicks this button when the data has been correctly entered to add the player to the table

Position Option Buttons
Here the user selects the position of the player to be used for the simulation