SpaceVis: NBA Court Spacing Visualizer

Kunal Shah
ks1300@scarletmail.rutgers.edu

Nicholas Kleene
nk374@scarletmail.rutgers.edu

***Abstract*— We propose a visualization system for the expected point value of a shot from any location on a basketball court, given the euclidean coordinates of each player on the court and the identity of each player. Our algorithm uses the shooting percentage by distance from the basket, shooting percentage by closest defender position, and the amount the defender reduces the opponent's shooting percentage to compute the expected point value of a shot from any location on the court. This approach uses publicly available data scraped from stats.nba.com to build a SQL database. The database is queried based on the player identities and euclidean coordinates that are input by the user to compute the heat map. This project has value for fans, journalists, and basketball teams. It improves on the current state-of-the-art visualizations (heat maps of player shooting percentages or frequencies), by providing visualizations by conditioning on extra variables (e.g. defender proximity, defender identity) and actually computing the expected point value to provide a more thorough understanding of player value.

I. Project Description

The goal of this project is to determine which basketball players improve their team's likelihood of scoring points. For a given floor configuration we will generate a heat map specifying the expected number of points for shots taken at each location on the court. This project is most closely related to the topic of prediction and verification. Currently, there are no widely available visual analytics that provide this information. The state-of-the-art for publicly available visualizations are heat maps that show the frequency with which a player shoots from each location on the court. This tool will allow fans, journalists, and teams to determine which players improve their team's chances of scoring, which enhances the understanding of fans and journalists and can allow teams to make lineup decisions based on this information. The project has three components that need to be completed: scraping stats.nba.com for data, programming the algorithm to produce the heat map, and creating the user interface. By week 6 we hope to have scraped all the relevant data and finished the algorithm. By week 9 we hope to finish the user interface. Finally, by week 10 these algorithm and interface will be combined to finish the prototype.

The project has four stages: Gathering Data, Design, Infrastructure Implementation, and User Interface.

A. Stage1 - The Requirement Gathering Stage.

- The general system description: The goal of this project is to generate heat maps that represent the expected points from a shot taken anywhere on the court, given a particular floor configuration. There are three components to this project. The first is actually obtaining the data. All of the data is publicly available at stats.nba.com, but needs to be scraped. The data is scraped and stored in CSV files based on different criteria like defender location, shooting percentage by location, etc. The data is at rest, however it is updated daily, as Once our database is created we will program the scraping to occur every day, in order to keep our database up to date. The second component is the algorithm for generating the heat map. Given a particular floor configuration, we need to compute the expected points from a shot taken from anywhere on the court. Finally, we need to create the user interface so the user can input player names and locations so we can generate the visualization.

  Users can interact with our system in two ways:

- The three types of users (grouped by their data access/update rights): There are three main types of users: basketball fans, basketball journalists, and basketball teams/coaching staffs. Basketball fans will use our system to help generate a better understanding of how particular players improve their team's chance of scoring. Basketball journalists can use our visualization to support their explanations in written pieces. Basketball teams and coaching staffs can use insights gained from these visualizations to teach players what they should do in certain situations to improve their team's ability to score. All three user types have will have read access but not write access.
- The user's interaction modes: Users can input floor configurations in one of two ways. In the first way users simply upload a text file with the players names, their locations on the court (in euclidean coordinates), and whether they are on offense or defense. Alternatively, users can use drop-down menus (for selecting player names and whether they are on offense or defense – string data types) and text boxes (for the location coordinates – float data types) to input this information player by player. Once this data is input, the system will generate the heat map.
- The real world scenarios:
  - Scenario1 description: User wants to determine who should take a shot, or where players should move to improve their team's shot, at a given moment in time for a given floor configuration.
  - System Data Input for Scenario1: Users need to provide the player's names, locations on the court

(in euclidean coordinates), and whether they are on offense or defense.

- Input Data Types for Scenario1: The data types are strings (player names and offense/defense) and floats (locations on the court).
- System Data Output for Scenario1: The output data is a heatmap visualization that shows the expected points for a shot taken at any spot on the court.
- Output Data Types for Scenario1: The output data is a matrix of floats representing the expected points for a shot taken at any spot on the court.
- Scenario2 description: User wants to determine whether a player improved his team's overall expected scoring output based on where he moved.
- System Data Input for Scenario2: Users need to provide the same set of data as in Scenario1, but they need two sets of location coordinates now (still only one set of players names and offense/defense designations).
- Input Data Types for Scenario2: The data types are strings (player names and offense/defense) and floats (locations on the court).
- System Data Output for Scenario2: The output data are two heatmap visualizations that shows the expected points for a shot taken at any spot on the court, given the two different floor configurations.
- Output Data Types for Scenario2: The output data are two matrices of floats representing the expected points for a shot taken at any spot on the court.

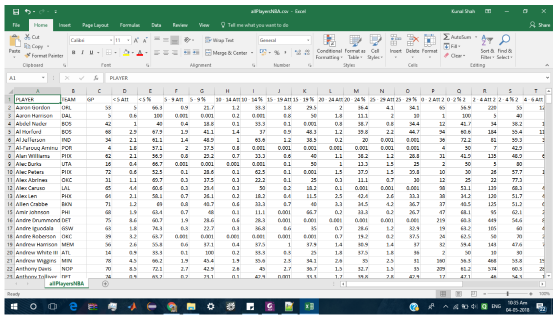
B. *Stage2 - The Design Stage.*

- A textual description of the flow diagram: Initially, the drop down will contain all player names in an alphabetical order. These names will be populated dynamically from the CSV file and will update if the file is updated. The user will select the players, select where they want them to be by clicking on the court, and choose whether they are on offense/defense. Once all the 10 players are selected, the algorithm will take those players and based on the statistics that we have scraped, and a heat map will be generated based on the statistics.
- The design flow: Initially, data will be scraped using Python. The front end will be designed using Bootstrap, HTML and CSS. The user will interact with the drop-down to select player names, click on the court locations to place players, and will select if a player is on offense/defense through a check box. Once all players are selected, the data will be passed to the algorithm, which will compute the output based on the player statistics. The output will be a heat map.

C. *Stage3 - The Implementation Stage.*

- Gathering player data and the back-end: The initial stage was scraping the data which would be the input to the algorithm. For that, We used the Selenium library of Python. Selenium automates browser clicks, and since our data was on different pages of www.NBA.com, it was required. We stored all the data in CSV files after scraping it based on various conditions as per requirement.
- Designing the front-end: Once we scraped the data, we designed the front-end using HTML, CSS and Bootstrap 4.0. We also used Jquery and JavaScript to make it dynamic. To develop the court, we used the grid system, a feature of bootstrap, to create a 470*500 grid, which is the size of the court. The players can click anywhere on the court to place a player on the court.
- Linking the CSV and front-end: To populate the drop down with player names so that the users can select them, we used the Django framework in Python, which creates a model, takes data from the CSV, creates a View and displays data. The Django framework would link back end to the front end. The output will be in the JSON Format, and using JavaScript, we converted the JSON to text which would be added to the drop-down list.
- Computing and Drawing the Heat Map: To compute the heat map, the player information that the user enters is parsed from JSON to text and passed to a JavaScript algorithm that computes the heat map. The result of the algorithm determines the color value in a 5x5 grid around each player. The heat map is then drawn using D3.js.

- Sample small data snippet



allPlayersNBA.csv - Excel

A1 PLAYER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	PLAYER	TEAM	GP	< 5 Att	< 5 %	5 - 9 Att	5 - 9 %	10 - 14 Att	10 - 14 %	15 - 19 Att	15 - 19 %	20 - 24 Att	20 - 24 %	25 - 29 Att	25 - 29 %	0 - 2 Att 2	0 - 2 % 2	2 - 4 Att 2	2 - 4 % 2	4 - 6 Att
2	Aaron Gordon	ORL	53	5	66.3	0.9	21.7	1.2	33.3	1.8	29.5	2	36.4	4.1	34.1	65	56.9	220	55	12
3	Aaron Harrison	DAL	5	0.6	100	0.001	0.001	0.2	0.001	0.8	50	1.8	11.1	2	10	1	100	5	40	
4	Abdel Nader	BOS	42	1	40	0.4	18.8	0.1	33.3	0.1	0.001	0.8	38.7	0.8	34.4	12	41.7	34	38.2	1
5	Al Horford	BOS	68	2.9	67.9	1.9	41.1	1.4	37	0.9	48.3	1.2	39.8	2.2	44.7	94	60.6	184	55.4	11
6	Al Jefferson	IND	34	2.1	61.1	1.4	48.9	1	63.6	1.2	38.5	0.2	20	0.001	0.001	36	72.2	81	59.3	3
7	Al-Farouq Aminu	POR	4	1.8	57.1	2	37.5	0.8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	4	50	7	42.9	
8	Alan Williams	PHX	62	2.1	56.9	0.8	29.2	0.7	33.3	0.6	40	1.1	38.2	1.2	28.8	31	41.9	135	48.9	6
9	Alec Burks	UTA	16	0.4	66.7	0.001	0.001	0.001	0.001	0.1	50	1	13.3	1.5	25	2	50	5	80	
10	Alec Peters	PHX	72	0.6	52.5	0.1	28.6	0.1	62.5	0.1	0.001	1.5	37.9	1.5	39.8	10	30	26	57.7	1
11	Alex Abrines	OKC	31	1.1	69.7	0.3	37.5	0.3	22.2	0.1	25	0.3	11.1	0.7	30	12	25	22	77.3	
12	Alex Caruso	LAL	65	4.4	60.6	0.3	29.4	0.3	50	0.2	18.2	0.1	0.001	0.001	0.001	98	53.1	139	68.3	4
13	Alex Len	PHX	64	2.1	58.1	0.7	26.1	0.2	18.2	0.4	11.5	2.5	42.4	2.6	33.3	38	34.2	120	51.7	4
14	Allen Crabbe	BKN	71	1.2	69	0.8	40.7	0.6	33.3	0.7	40	3.3	34.5	4.2	36.7	37	40.5	125	51.2	6
15	Amir Johnson	PHI	68	1.9	63.4	0.7	48	0.1	11.1	0.001	66.7	0.2	33.3	0.2	26.7	47	68.1	95	62.1	2
16	Andre Drummond	DET	75	8.6	60.7	1.9	28.6	0.6	28.3	0.001	0.001	0.001	0.001	0.001	0.001	219	60.3	449	54.6	8
17	Andre Iguodala	GSW	63	1.8	74.3	0.3	22.7	0.3	36.8	0.6	35	0.7	28.6	1.2	32.9	19	63.2	105	60	4
18	Andre Roberson	OKC	39	3.2	63.7	0.001	0.001	0.001	0.001	0.001	0.001	0.7	19.2	0.2	37.5	24	62.5	50	70	2
19	Andrew Harrison	MEM	56	2.6	55.8	0.6	37.1	0.4	37.5	1	37.9	1.4	30.9	1.4	37	32	59.4	143	47.6	7
20	Andrew White III	ATL	14	0.9	33.3	0.1	100	0.2	33.3	0.3	25	1.8	37.5	1.8	36	2	50	10	30	
21	Andrew Wiggins	MIN	78	4.5	66.2	1.9	45.4	1.9	35.6	2.3	34.1	2.6	35	2.5	31	160	56.3	468	53.8	19
22	Anthony Davis	NOP	70	8.5	72.1	2.7	42.9	2.6	45	2.7	36.7	1.5	32.7	1.5	35	209	61.2	574	60.3	28
23	Anthony Tolliver	DET	74	0.9	63.2	0.2	23.1	0.1	42.9	0.001	33.3	1.7	39.8	2.8	42.9	17	47.1	46	54.3	1

allPlayersNBA

Ready

10:35 Am 04-05-2018

Fig. 1. Sample small data snippet

- Sample output snippet
-
- - Data The data consists of about 500 players (only 62kb), and it is not streaming data. It is disk-resident. This consists of about 95 percent of the current NBA players.
  - Findings The quality of a team's shot (average expected point value) is heavily dependent on who the players are and where they are located on the court. For example, LeBron James has a very high expected point value when he is near the basket, but much lower when he is far away. As another example, Stephon Curry has a very high expected point value anywhere on the court. The identity of the defender,

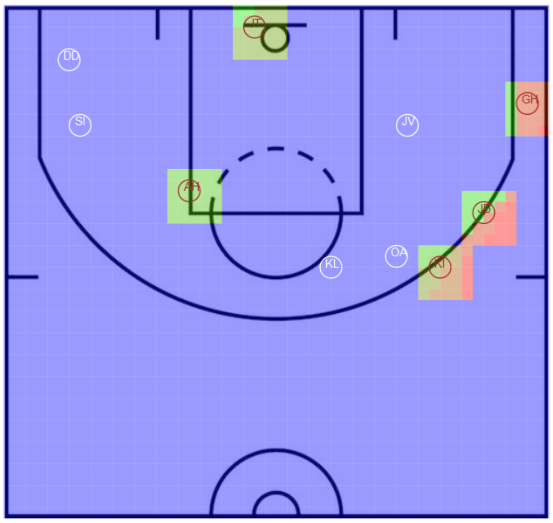


Fig. 2. Sample output snippet

on the other hand, has a very small influence on the expected point value for each player. Instead, it is more important how a defender's proximity changes the offensive player's probability of making the shot. Finally, the best expected point values occur when there are many players behind the 3-point line. This is because the maximum point value of a 3-point shot is higher than a 2-point shot. This reflects the current style of basketball being played in the NBA now where the best teams (e.g. the Houston Rockets) take a high number of 3-point shots. Therefore, the best offensive lineups are ones that feature many capable 3-point shooters.

– End goal: There is a lot of data produced by the in-arena cameras, however, the techniques for visualizing the NBA data have lagged behind. The heat map we generate will give the teams, journalists, and fans a better insight into how much each player contributes into winning.

D. Stage4 - The User Interface.

The user interface consists of a drop-down menu, two check boxes, and a grid of transparent squares overlaid on the image of half an NBA court. The drop-down menu contains player names, the check boxes allow for designating players as offense or defense, and the grid is used to select the locations of the players on the court. After each player is input, a circle with their initials is drawn on the grid. After making these selections for 10 players (5 on offense, 5 on defense), a heat map is generated.

- Mode of Interaction: The user selects player names, offense/defense designations and court locations using mouse clicks.
- Error Messages: If the user tries to pick a court location without selecting a name or offense/defense designation the user receives a message asking them to pick select a player (or offense/defense designation). If the user tries to input more than 10 players, they are informed that they can't add any more players. If the user tries to input more than 5 players on offense or defense, they are informed they can't add any more players with that designation.
- Feedback to User: Mousing over court locations changes the color to show the user where they are. As each player is entered into the court, a circle with their initials is drawn at the selected location. After entering all 10 players, a heatmap is generated (with circles and initials corresponding to player locations). The average expected point value is also displayed.
- Data Range Violations: There are no error messages related to data range constraints. The user is only allowed to select players from the drop-down menu (only players we have data on will appear in the drop-down menu). Additionally, the user can't actually input the player without selecting the court locations. The one limitation is that our data only goes to locations 29 feet from the basketball hoop, so any heat map locations greater than 29 feet from the hoop are automatically set to 0.
- Interface Mechanisms: The interface mechanism used for data access is just the drop-down menu, the check boxes and the court for entering player locations. The player locations are updated as they are added. When the user enters the information for all 10 players, the heat map is drawn and the average expected point value is displayed to the user.

Please insert your deliverables for Stage4 as follows:

- Sample Navigation Path

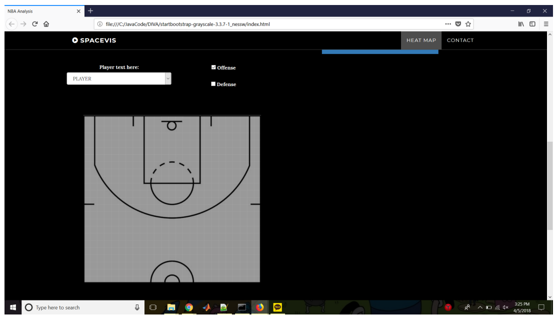


Fig. 3. Initial User Interface

- Error Messages:
  – Error Message 1: Please Select a Player. Triggers when the user selects a court location before selecting a player.
  – Error Message 2: Please Check Offense/Defense. Triggers when the user selects a court location before selecting a offense/defense designation.
  – Error Message 3: Too Many Players on Offense. Triggers when the user tries to input more than 5 players on offense.

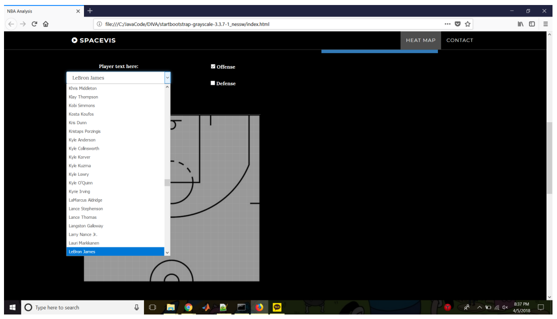


Fig. 4. Player Selection

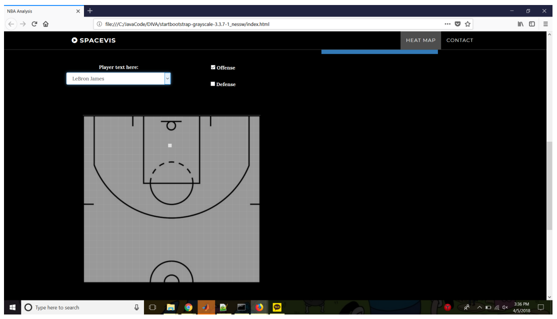


Fig. 5. Selecting Court Locations

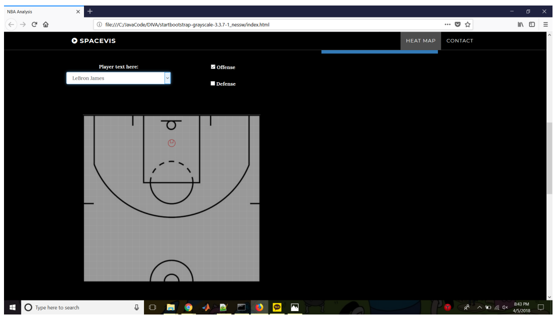


Fig. 6. UI after one player has been input on offense

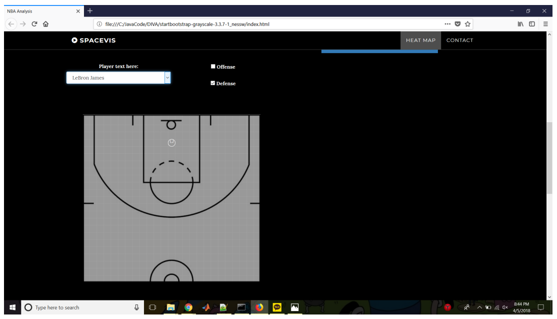
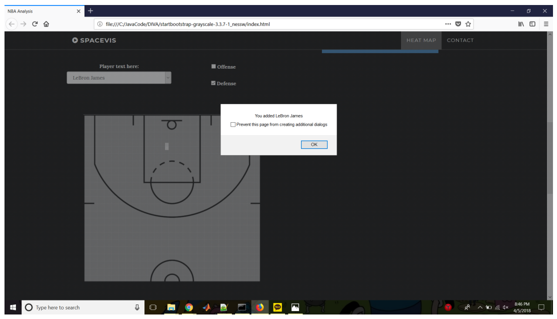


Fig. 7. UI after one player has been input on defense

- Error Message 4: Too Many Players on Defense. Triggers when the user tries to input more than 5 players on defense.

- Information message: Average Expected Point Value = X. This message triggers when the user inputs 5 players on offense and 5 players on defense. There are no data range constraints since the user is not allowed to select anything out of range to begin with.
- Interface mechanism: The user selects 5 players for offense and 5 players for defense, selects their locations on the court, and then the heat map is generated.


You added LeBron James

Prevent this page from creating additional dialogs

Fig. 8. UI after message after adding a player

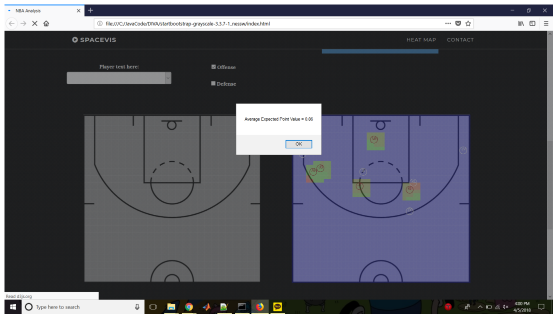

Average Expected Point Value = 0.86

Fig. 9. Resulting heatmap output (displaying average expected point value)

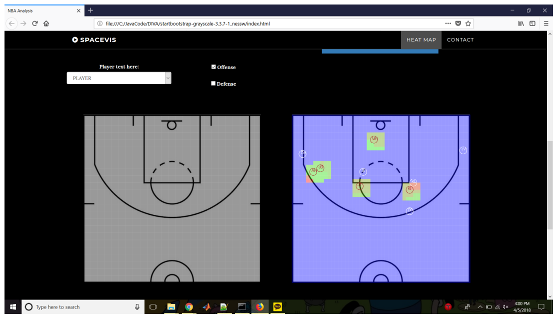


Fig. 10. Resulting heatmap output

References

[1] Ernst, M. O. & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. Nature, 415, 429-433.

[2] Heer, J., Bostock, M. & Ogievetsky, V. (2011). D3.js [software]. Available from https://d3js.org/

[3] Holovaty, A. & Willison, S. (2005). Django [software] Available from https://docs.djangoproject.com/en/2.0/intro/tutorial01/

[4] Holovaty, A. & Willison, S. (2005). Django [software] Available from https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/

[5] Otto, M. & Thronton, J. (2011). Bootstrap [software]. Available from https://getbootstrap.com/docs/4.0/getting-started/introduction/

[6] NBA Media Ventures (2018). NBA statistics obtained from http://stats.nba.com/