Capstone Project Report

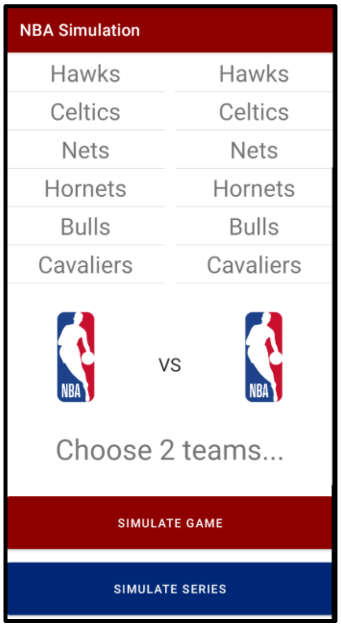
Carrigan Holt

Introduction:

Being a livid basketball fan and obsessed with video games, it is no surprise I play a lot of NBA 2K. The creators of NBA 2K pride themselves on 2K being the most realistic basketball video game on the market, but I had discovered an issue with this claim. The games simulated are very unrealistic from the actual games. I decided I wanted to put their algorithm to the test. For my project, I wanted to create a basketball simulation more accurate than NBA 2K's simulation. I will also compare it to Vegas betting lines to see how accurate I can make it.

Objectives:

- Simple, easy-to-use user interface (Android app).



- Create an algorithm to simulate each game.

  The algorithm works as follows:

  1. Who has the ball?
  2. Does that team turn the ball over or get the ball stolen?
  3. Where does the team take the next shot from?
  4. Are they fouled when they take the shot?
  5. Do they make the shot or get their shot blocked?
  6. Do they make their free throws (if any)?
  7. Which team gets the rebound?
  8. Repeat until the game reaches the calculated number of possessions.

- Research popular simulation algorithms.

  NBA 2K uses individual player tendencies to simulate their games. For example, if you put Player X on the Utah Jazz, he would still play exactly like he played on his former team. When in real life, Player X would most likely adjust to how the Utah Jazz play, meaning his tendencies would change. I took this into consideration by basing my simulation of team-wide tendencies.

- Test accuracy of simulations against real results.

  Using the team statistics from the 2020-21 NBA season (before the COVID-19 shutdown), NBA 2K20 (PC), and the Vegas betting odds for each game, I was able to determine how accurate each source is at predicting NBA games. Here are the results:

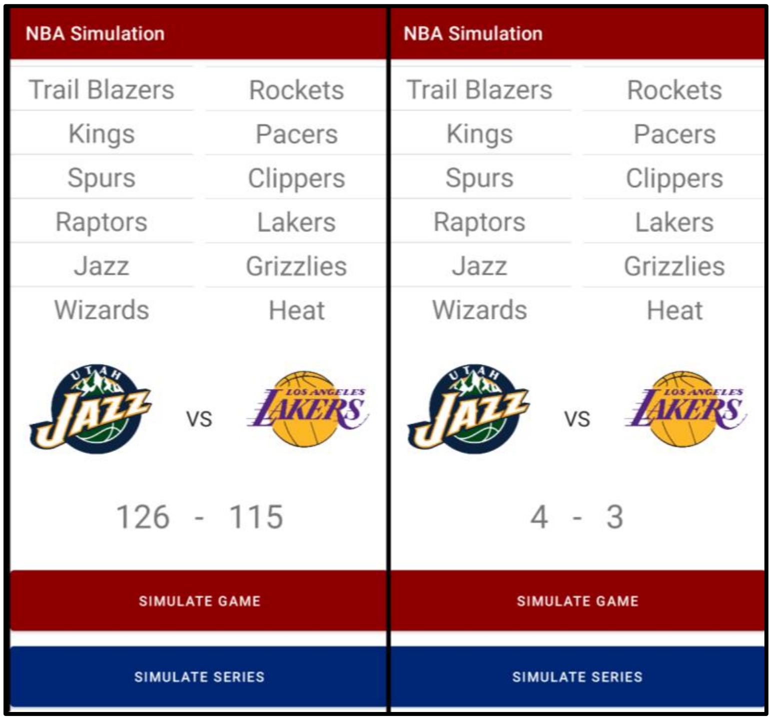
	Actual	2K20	Betting Lines	My App
Record	41-23	25-39	54-10	40-24
Correct	-	45%	67%	58%

As you can see, my app was more accurate than 2K's simulation, and very close to the accuracy of betting lines. One issue that comes into play when predicting the outcome of sporting events is luck. Buzzer beaters, breakout games, bad officiating, etc. all affect the outcome of games but are almost impossible to predict the impact of them. To adjust for this, we need to look at the data again. Instead of looking at each game individually, I looked at the series results between 2 given teams. For example, the Utah Jazz played the Minnesota Timberwolves 3 times, and the Jazz won all 3 times. If a simulation only had the Jazz winning 2 times, I only gave that simulation 2 out of 3 points. Using this, we have an adjusted accuracy for each simulation:

	Actual	2K20	Betting Lines	My App
Correct	-	55%	73%	70%

This boosted all the scores, but 2K still fell way below my app's and the betting line's accuracy. The betting lines consider injuries, story lines, and other things not represented in the statistics, so it makes sense that those are more accurate than my app, but my app is not far behind, being only 3% less accurate. I want to keep working towards making my app more accurate than the betting lines, but I am very happy with how it has turned out so far.

- Present results in an easy-to-understand report.



126 - 115

4 - 3

I minimized the output so users can more easily understand what the app has calculated. I wanted just a single line of output, the score of the game or series selected by the user. I wanted this to be an application that anyone could use, not just those who dive deep into basketball analytics.

- Include all NBA teams preloaded into the application.

All NBA teams are included in the app, and all statistics are updated as of 4/18/2021. One challenge I encountered early on was wanting to have the stats automatically update whenever the program was run or the app was opened, but this proved difficult with the specific stats I ended up using.

- Provide an option to predict a series instead of a single game (best 4 out of 7).

  I added 2 buttons for simulations, one to simulate a single game and output the score, and one to simulate a series and output the series score. This will be useful when people start wanting to predict the outcome of playoff matchups.

- Include team logos in application for a better user experience.

  Team logo .PNG files were used within the app to minimize user error and create a more fun experience for the user.

- Interface animations for a better user experience.

  Android does a good job of adding animations to its interface, such as scrolling lists and button animations. Once I decided to switch to an Android app instead of a Java application, it greatly enhanced the user experience and portability of the tool.

- Include NCAA teams as well as NBA teams.

  Although this was not actually implemented, it would be as easy as generating a new text file with the NCAA statistics and uploading the team logos into the app.

I made the app so it would be very easy to adjust team stats or add whole new teams. You just need a logo image and the required team statistics.

Literary Review:

- "Locked on Jazz" Podcast

  I listen to this podcast every day, and it was my greatest resource when designing my algorithm. The host, David Locke, really emphasizes the importance of a team's shot locations and how big of a difference it can make to force other teams out of their usual shot profile. For example, a team that shoots mostly 3-pointers at 40% accuracy, they get on average 1.2 points per possession. To match that point production, a team that doesn't shoot 3's would have to make their 2-point shots at 60% accuracy. Looking at where team's take and make shots gives good insight to how that team will perform. I chose to base my algorithm on shot charts due to how important they are to a team's success.

- "KOT4Q" YouTube Channel

  Kenny is one of the YouTube leaders when it comes to exploring 2K's simulation. A lot of his videos are silly, but a few of them dive deep into what is wrong with 2K's simulation, how it works, and how to break it. I used his insights to avoid common errors found in 2K to better improve my own application. One of the biggest complaints about 2K's simulation is the time at which it takes 2K to simulate a game. It can take a few seconds, up to a minute, to simulate a single game in 2K. I really wanted to make sure speed was a high priority in my application.

- "Mobile App Development for Android" Class

  I am taking this class this semester, and it is what inspired me to venture outside of Java when I was not liking the UI/UX. Dr. Hussain Aljafer has been amazing helping me discover features within Android Studio to help me with this project. As I learn more about Android applications, I hope to add even more features.

Lessons Learned:

I learned a lot by completing this project. Not only about specific coding utilities, but about the entire process of completing the project. I gained the skills to create an Android app from start to finish, as well as how to manage a project timeline. I also learned that choosing a user interface that best suits your project before you build the project is extremely important and can save you a lot of time. Next time, I will be more thorough when selecting the platform users will be interacting with.

Conclusion:

I was able to complete all my objectives. The biggest obstacle I had to overcome was the Java interface not looking as modern as I would have liked. To combat this, I decided to import the algorithm into an Android application. This took some extra work and had a much steeper learning curve, but in the end, I think it paid off. The UI/UX is much sleeker and easier to use, as well as giving the utility the ability to fit in your pocket. As I learn more, test the app more, and think of new features to add, I eventually want to put this app on the Google Play Store and give everyone access to this tool.

Submissibles:

- Source Code
- Project Report
- Presentation

Appendices:

- Source Code: NBASimulationApp.zip
- Report: NBASimulationAppReport.pdf
- Presentation: NBASimulationApp.pptx