Overview and Motivation: Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.

We're both really into playing League of Legends, and we're very excited about the new game mode, Ultra Rapid Fire, that was released on April Fools Day. This new game mode has brought about a lot of changes, and we would like to use visualizations to further explore those changes in a way that no website or service currently allows.

One of the most important things in a game like League of Legends is game balance. There are 123 champions to choose from, and the game developers, Riot Games, do their best to make sure no one champion is categorically stronger than any other. Normal game modes have been around for years, so Riot has had plenty of time to make sure that all champions and items are as balanced as possible in normal. However, since this game mode is so new, there are drastic balance issues that we are excited to learn more about.

Related Work: Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.

This project is something we were interested in doing since the very beginning of the class, so there is no one clear motivation for the project other than our interest in the game. However, there are some websites that currently assess the current balance/win rates of champions in the game, as well as popularity and win rates of item builds, although none use particularly compelling visualizations. Some of these websites/visualizations can be seen here:

- http://www.metalol.net/champions/
- http://www.op.gg/statistics/champion/

However, these websites all focus on statistics gathered from games played in normal game modes, not Ultra Rapid Fire mode, since Ultra Rapid Fire mode is such a brief phenomenon. This means that our project will answer questions that not many other websites/developers are trying to solve.

We were also impressed and partially inspired by last year's Data Driven Dota project. Although our project will be much different because it will assess fairness and balance on a general level instead of looking at statistics for specific users, some of the elements of their visualization would be very useful in our visualization.

Questions: What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?

We would like to explore exactly how the game of League of Legends has changed due to the introduction of the Ultra Rapid Fire Game mode. Riot Games puts in a lot of effort making sure champions and items are balanced and fair in normal games, but URF disrupts that balance.

We want to be able to explore the popularity and effectiveness of certain champions, popularity and effectiveness of items, changes in game pace, etc. in this new game mode. The visualization we create will give users an idea of how to compose the strongest team with the best item builds, find sleeper picks, and avoid picking champions that don't live up to their hype.

Some of the questions that we hope will be answered by our visualization include the following:

- Which champions are the most popular in URF?
- Which champion should I pick to give my team the greatest chance to win an URF game?
- Given the champion I want to play, which items should I buy, and which should I stay away from in order to maximize my chances of winning the match?
- Which champion choices result in the shortest game durations? Is this because they are very good, or because they are very bad?
- How are metrics like kills, deaths, and assists affected by champion selection?

Data: Source, scraping method, cleanup, etc.

Riot Games (the developer of League of Legends) has an API to pull data for specific games. With the introduction of Ultra Rapid Fire mode, they also introduced a function that, given a specified time, would return a randomized set of game ids of games completed within five minutes before the specified time. This appears to be the only way to gather data for URF games, so we wrote a script that, for every five minute interval since the introduction of URF, called the function and saved the game ids that were returned. This resulted in a list of 35,000 games, which our script then exported as a csv file.

The Riot API also includes a function that, given a specific game id, returns data for that game. However, the data for each game is in a json of about 150 kB in size, so it would be impractical to save all that data and use it in our visualization. We wrote a script that, for each game id we had, used the Riot API function to pull the json containing all data for that game, then pulled specific fields to create a new json that could be used to construct the data for our visualization. We saved all of these shrunken-down jsons, each representing one game's worth of data.

Our final data processing script began with an empty json frame with space for 123 dicts, each of which represented all the metrics we were interested in for each champion. Then, for each game data json we had for all of our 35,000 games, we added data to the master json for each champion in that game to reflect each champion's performance in each game.

We noticed that 35,000 games worth of data resulted in a json that was quite large (87 MB), so we are currently considering using a smaller subset of games (around 1,000 games), which yields a master json file of only around 1.6 MB, which seems much more manageable for our project.

Exploratory Data Analysis: What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?

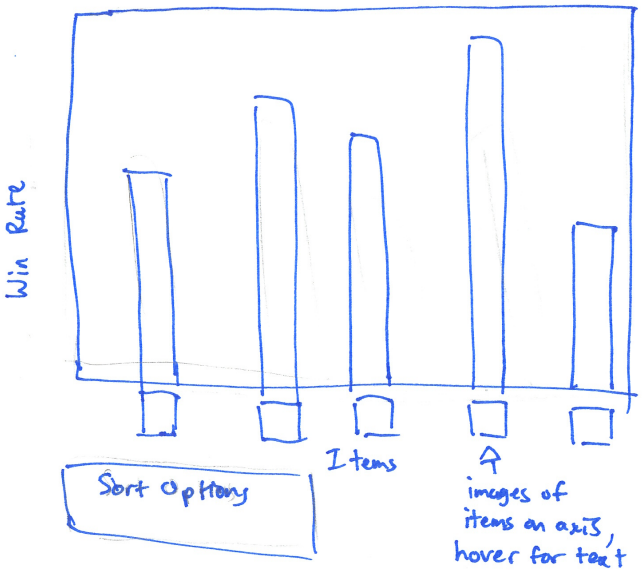
We initially considered using a central visualization that took the form of a bubble chart, in which each champion was displayed as a bubble. The y-axis was average win-rate of the champion, the x-axis was the average game duration for the champion, and the size of the bubble corresponded to the popularity of the champion. While this gave us an interesting chart and answered some of our main questions, it didn't lend itself well to being an interactive visualization, and was mostly static. This showed us that we needed to come up with a design that would

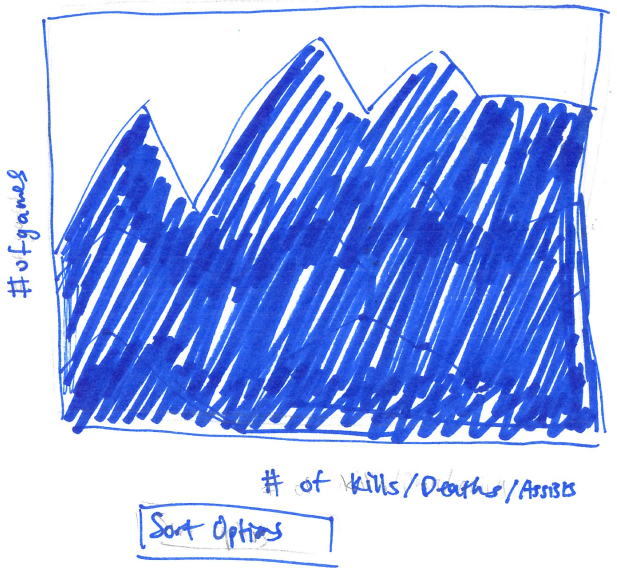
be very dynamic as we clicked on different champions and other various elements of our visualization and led us to the direction we are now planning to take.

Design Evolution: What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course.

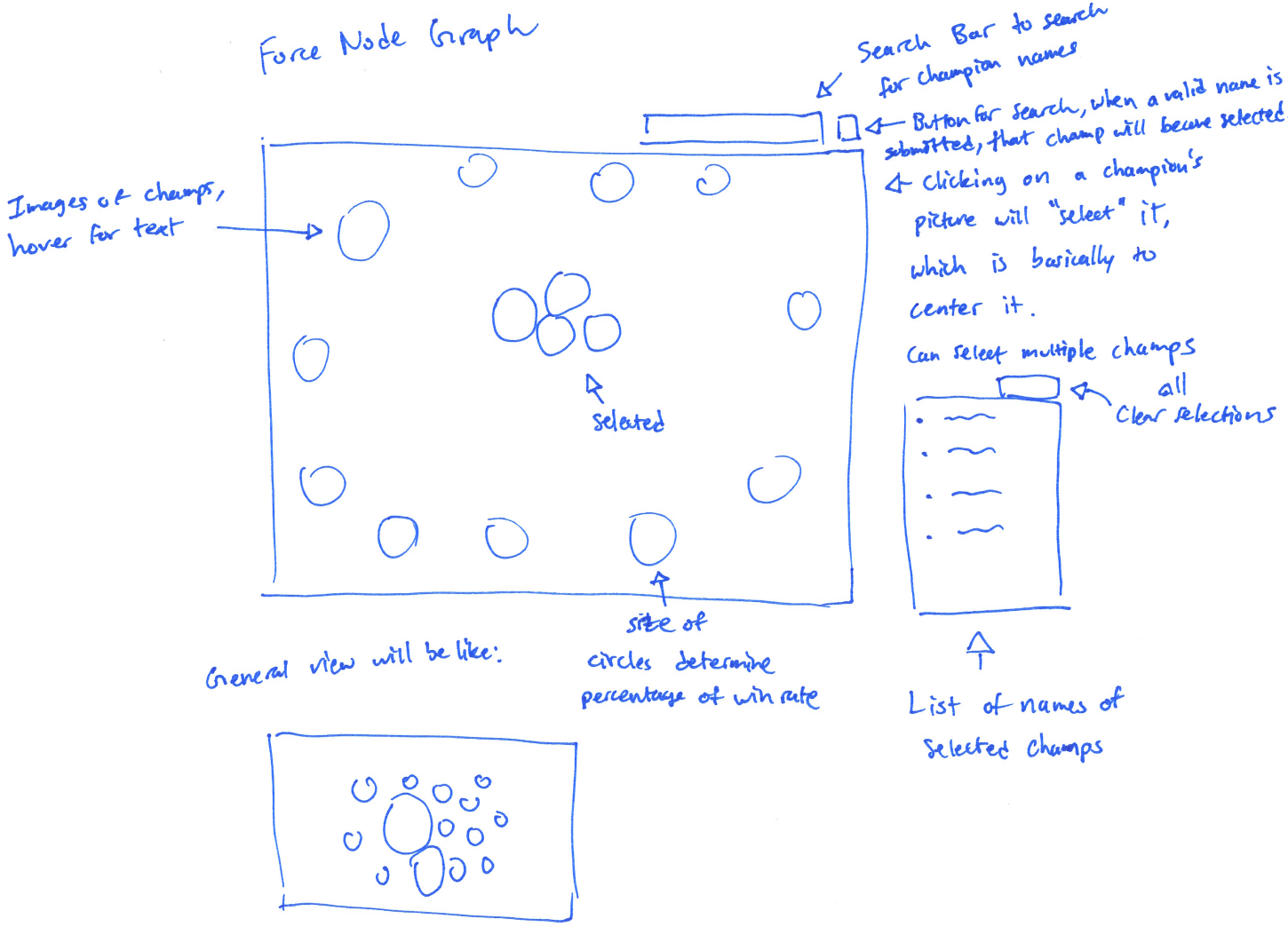
We were originally going to implement a bubble chart that was static and represented each champions win/lose rate. However, we quickly realized that this might cause problems since it was possible that bubbles would overlap. To fix this problem, we decided instead to change the bubble chart to a force layout chart. This way, overlapping wouldn't be a problem, and we could change this chart to become a cool display for filtered champions at the same time.

Explanation of graphs here.





Force Node Graph



Implementation: Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

As of now, the major interactions occur with the force node graph as well as the area graph that show the number of games that occurred with a specific game duration. See images for how the interactions work.

Evaluation: What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?
TBD