Data Driven Dota, Process Book for CS171, Spring 2014

Angela Fan, Robbie Gibson, Benjy Levin

April 2014

1 Overview and Motivation

Dota 2 is a multiplayer online battle arena game developed by Valve Corporation. Players play individual, independent matches that each take about an hour. Matches involve two teams of five players, one side called "Radiant" and the other side called "Dire." Players each control a character called a "hero," which starts off weak, and grows in strength through leveling up and gaining gold. Levels allow players to unlock or upgrade unique abilities, while gold allows players to buy items that improve their hero. Players play on a square map (please see figure), with the lower left hand corner and upper right hand corner occupying a stronghold for the Radiant and Dire, respectively. The game ends when the opposing team's stronghold, called an "Ancient," is destroyed. These Ancients are protected by defensive structures called "Towers," that the opposing team can destroy for a gold bonus. Please see Figure 1 for an image of the Dota2 map.

Figure 1: Dota Map. At each end of the map are the Radiant and Dire strongholds, called Ancients (marked by a rectangle). Defending the Ancients are a series of Towers, drawn as circles.

Dota2 players, however, do not have many forms of secondary analysis with which to critique their play, and help them improve it. The primary source of Dota2 statistics is the website Dotabuff, which provides some summary statistics for each player, information related to individual matches, and trend information for heroes and items over the entire playing population. However, we wanted to implement a site whose central theme is time—we wanted to allow users to explore their entire match history, glean information about their play style and habits, and see trends in their gameplay.

2 Related Work

As mentioned above the current Dota 2 statistics websites are aimed at a different niche. Dotabuff does an excellent job of letting you see any one match from your entire history and it even has a few summary statistics. It also examines

trends over all public games ever played. However, it doesn't have very many individual statistics or a way to examine your play on a higher level than just looking at all your matches.

The other major Dota 2 stats website is Datdota. However, this aggregates statistics from professional games only, and is thus useless to a player wanting to see where they personally could improve.

3 Main Questions

As a user...

1. How can I compare how I played in different games?
2. How can I see if I improved over time?
3. How can I see if my game strategy was effective?
4. How can I tell what heroes I play best?

4 Data

4.1 Gathering Data

The important data we are visualizing is the history of a given user's matches. To get this data, we used the Valve API to query for a user's data. Unfortunately, the API had some major problems, although we was able to work around most of them. The biggest problem for us, not using a real web server, was that the API didn't support cross-orign resource sharing. In a perfect world, we'd figure out how to run a web server and do the querying from a script somewhere, rather than client-side Javascript. Furthermore, as Valve requested we only make approximately one API call per second, and players can have in the thousands of matches, this further limited our ability to dynamically gather data for any user. Again, in a perfect world, we'd store our information in a database, only querying the API to update things, but that's also difficult, and not the best use of our time on this project. In the end, we used Python to statically gather the matches of 6 users (also, we ended up only being able to gather the most recent 500 matches for any user) and then save that for use.

We also had to gather a bunch of supplementary data for display purposes. The Valve API returns things like heroes played and items bought in the form of unique numerical ids, so we needed a way to convert those ids into actual names and information. For this, we built a JSON object that mapped ids to all the useful data about heroes, items, and abilities. This information was acquired using information already assembled by other users and a series of undocumented further Dota 2 API calls.

We also had to gather all the images required for the project. Each hero, item, and ability has a recognizable picture in game, and displaying those in our visualization would be a major benefit. We decoded the format Valve used to store the images on their servers, and, used a Python script to pull the images and store them in our own files.

4.2 Interpreting Valve API Data

We felt that we should create our own library since we will be making many function calls that require the Valve API data across all of our mini-visualizations. This was particularly pertinent for the game meta data such as the names of heroes and their main attributes, item information and all game related images. We store all of this information in corresponding JSON files, and we only require to load this data into our visualization once from our library. Any subsequent requests of this information is made through public methods that are available via our library.

We named our library D2 - as a fun spinoff of Dota 2 and D3.

5 Design Studio Feedback

We received feedback from **Matthew Warshauer**.

He made a number of good points, which we enumerate below. His comments are written in boldface, our responses are below each comment.

1. **Is it possible to learn about aggregate behavior of other Dota players as a way to evaluate your own play?** First, the API calls are difficult to make, because Valve limits you to 1 call per second, and we would need a larger set of users, since you can only get the latest five hundred games per user. Assuming we got

this data, evaluating other players in a naive aggregate way is not ideal. This is because when you play a Dota game, you only play against people of similar skill level (based on Valve matchmaking), so a pure aggregation would not be successful in allowing you to learn how to get better. A very good player probably doesn't want to see aggregates based on a pool of players that may be primarily worse than s/he is. Additionally, the Valve ranking is private and cannot be retrieved from the API data, so we have no way of filtering which users are better players than others. We agree this is a good idea, but does not seem possible at this point.

2. **Is it possible to do dynamic API calls?** Yes, but it would take a long time. As we addressed above, making one API call takes one second, and each user's games must be queried individually. Ideally we would build a database and call all users matches, and then load data from the database, but that is a little beyond the scope of the project.

3. **Hero chord diagram is not good. Should replace with matrix diagram.** We believe that this is an excellent suggestion, and will look at the Les Mis co-occurence matrix for ideas. This is definitely a good way to visualize this kind of data, but it needs to be filterable. An issue though is that this type of visualization would have a low data to ink ratio, because much of the matrix may be empty (i.e. if we have ten heroes, they may play often with some heroes, but much of the 10x10 matrix may be blank). Additionally, such a matrix has the same problems as a genetic heatmap- often not very easy to read and hard to tell what to sort by.

   A possible other avenue of exploration, inspired by Matthew, is adding more functionality to our chord graph. We should have a description of the chord graph and what it is showing, tooltip over each chord so you can see the number of games those two heroes have played, and be able to read off what elements those two chords link.

   That being said, we will certainly pursue both ideas and describe which one we ultimately go with as a design decision.

4. **Color coding of bars by cost isn't very effective, since you already have sorting by cost.** We agree. We think we should color-code by win/loss rate with that item.

5. **Perhaps bring up a list of games in which you bought this item. Can you see if a certain item is correlated to winrate?** That seems possible, and would be very interesting on items that are not commonly purchased- for example, if I as a player run the risk of buying this rare costly item, do I win with it? Is there a very expensive item that even if I purchase, acts like wasted gold?

   However, this is difficult for very commonly purchased items. We will start implementing this idea and see if there is a number of games played cutoff for displaying the games. Additionally, we are not yet sure where to put this games list, but will work on that when we decide layout.

6. **Graph descriptions are necessary.** Yes. We have not yet made these, but the final product will have these.

7. **Everything that is filterable should be un-filterable. User should know where s/he can filter.** We agree. We will have a description in each graph and at the top of the page about the filters. We will try and make sure the filters go clearly with the correct graph, and that it is clear to the user what filters what. We may have a "help" button to help users navigate this system.

8. **Green means two things- win, and agility. That is confusing. I recommend red/black for loss/win, respectively.** We agree, to an extent. However, the shades of green are clearly different- agility is closer to a highlighter shade, while the winrate green is a dark green. Red/black for loss/win we do not think is as intuitive as red/green. We will look into arranging graphs with similar colors next to each other, so color is more of a visual link rather than a differentiator between graphs.

6 Exploratory Data Analysis

7 Design Evolution

8 Implementation

9 Evaluation

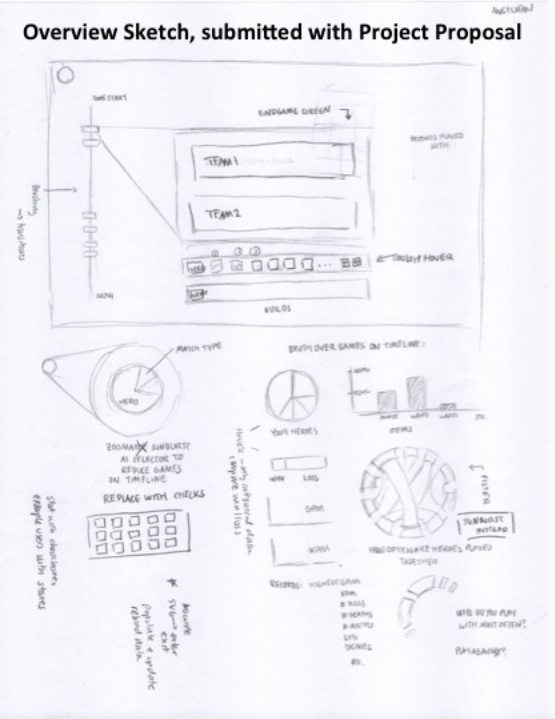


Figure 2: Design Process

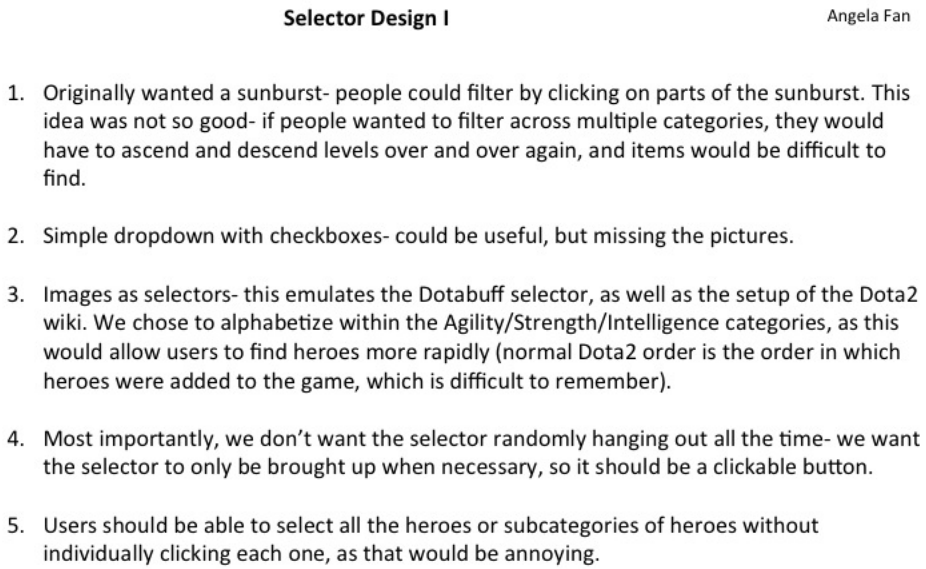


Figure 3: Design Process

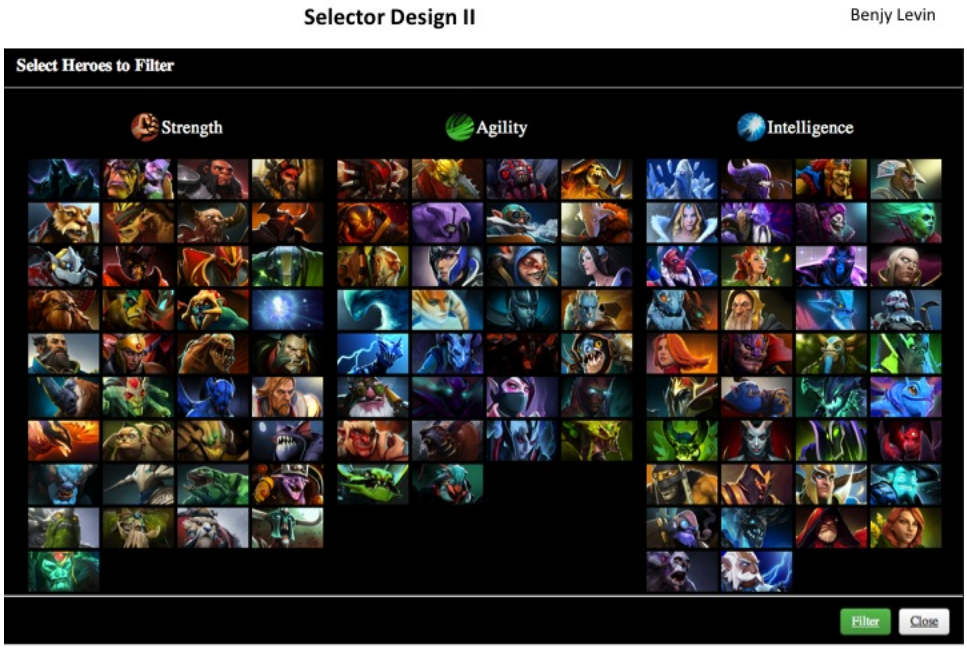


Figure 4: Design Process

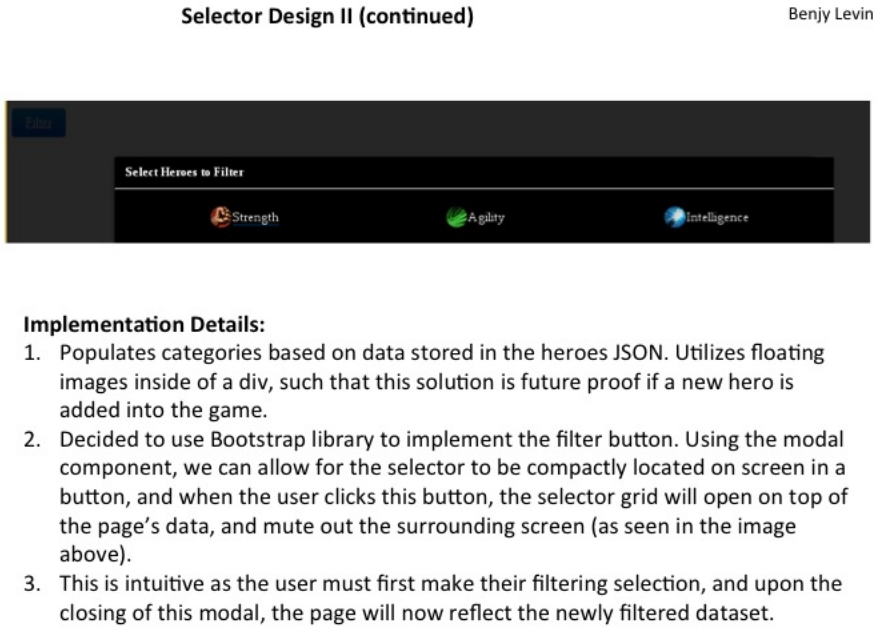


Figure 5: Design Process

Selector Work III Robbie Gibson

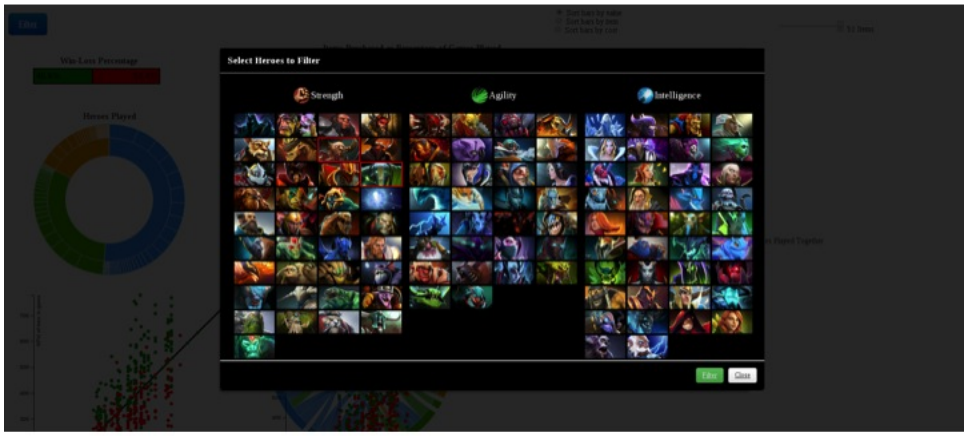


Figure 6: Design Process

Designing the Statistical Graphs- the Hero Pie Chart I Angela Fan

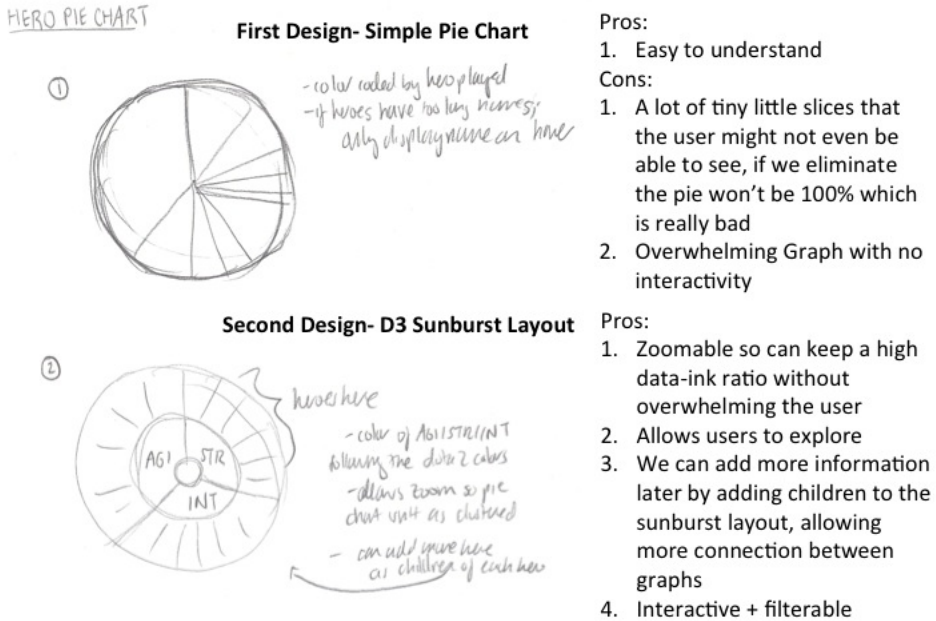


Figure 7: Design Process

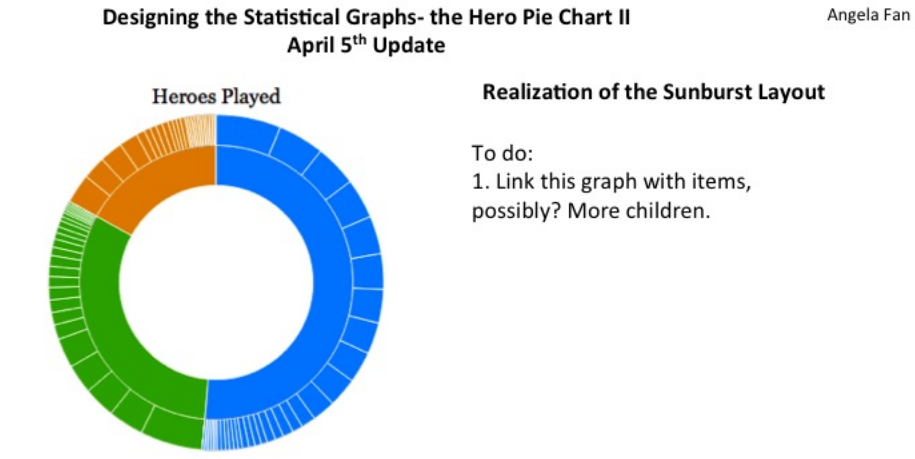


Figure 8: Design Process

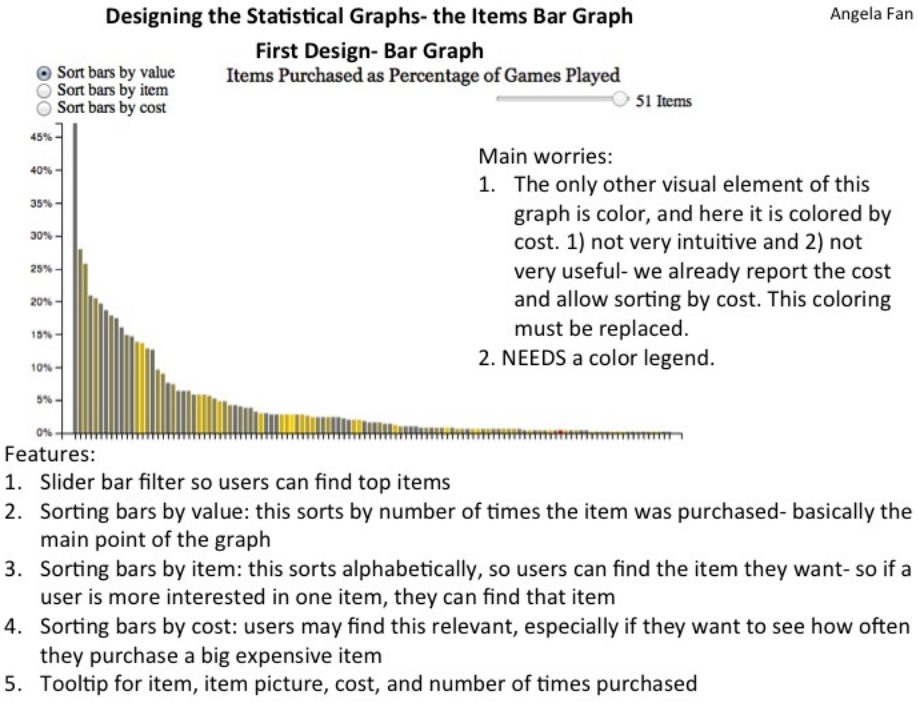
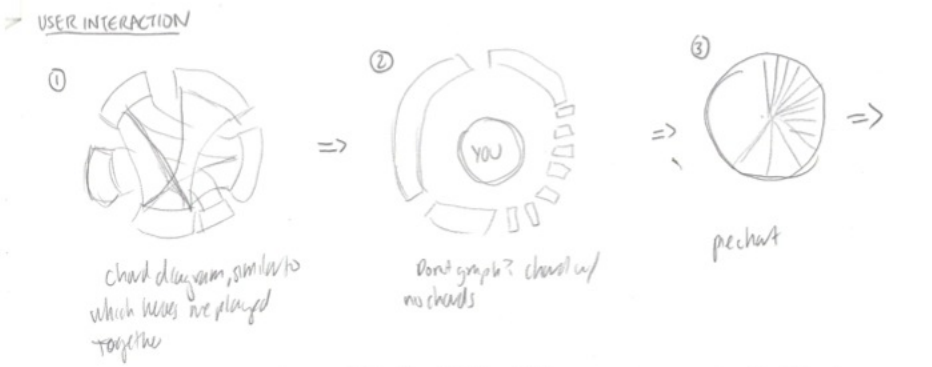


Figure 9: Design Process

Designing the Statistical Graphs- the User Interaction Graph, Part I Angela Fan



Chord Diagram

Pros:

1. Looks pretty cool, makes sense intuitively

Cons:

1. We don't have the data for that many users, so we can't represent interactions between other users. There aren't that many chords!

No chords Chord Diagram

Pros:

1. Makes sense. Chord diagram without chords.

Cons:

1. Chord diagram without chords == bar chart?

Pie Chart

Pros:

1. Most sense! Intuitive!

Cons:

1. Users who I've only played one game with aren't very interesting, but if we filter, the pie chart no longer represents 100% anymore

Figure 10: Design Process

Designing the Statistical Graphs- the User Interaction Graph, Part II Angela Fan

Bubble Chart

Pros:

1. More interesting visually than pie chart
2. Can filter it so it doesn't necessarily need to represent 100%, but can display only users that you've played more than one or two games with

Cons:

1. Not as intuitively understandable as a pie chart
2. The central bubble that represents "you" is very very large compared to the other bubbles

Figure 11: Design Process

Designing the Statistical Graphs- the GPM/XPM Graphs I Angela Fan

Line Graph

Time on X-axis, GPM on Y-axis
Pros:
1. Basic

Cons:
1. Not good measure- no normalization
2. No good comparative ability
3. Terrible way to detect trends

Line Graph II

Time on X-axis, GPM on Y-axis, but only the user's average is plotted. There is a dropdown menu at the top of the graph, so user can pick to change which hero is plotted

Pros:
1. Can now try and detect hero-based trends

Cons:
1. Not good measure- still no normalization
2. No good comparative ability between heroes
3. Introduces yet another selector that overcomplicates the whole graph system

Figure 12: Design Process

Designing the Statistical Graphs- the GPM/XPM Graphs II Angela Fan

Scatterplot I

Average GPM for that hero on the x-axis, GPM for that game on that hero on the x-axis

Pros:
1. Comparative! Not between heroes but at least between games

Cons:
1. Not intuitive, since below the line is a "good game"
2. Still no normalization, and not comparative between heroes

Scatterplot II

Average GPM for that hero on the y-axis, GPM for that game on that hero on the x-axis

Pros:
1. Comparative! Not between heroes but at least between games
2. Intuitive, since being above the line is great game and being below the line is a bad game

Cons:
1. Still no normalization, and not comparative between heroes

Figure 13: Design Process

Designing the Statistical Graphs- the GPM/XPM Graphs III, April 5 Update

Angela Fan

Normalization Needs

1. Over game length- in a longer game, heroes on average get much stronger than in a shorter game
2. Data bias- sometimes the game ends accidentally before it actually begins- these GPM/XPM values are skewing the mean
3. Based on hero role- if you play different roles, sometimes you have different GPM/XPM. If a hero can fit multiple roles (carry/support, like Alchemist), then the mean isn't a good representation of an average game of either role. For example, if you play carry Alch 50% of the time and support Alch 50% of the time, the mean GPM floats between the two, but that doesn't mean that your support Alch games are "worse" games on average
4. This graph doesn't help evaluate how heroes do with farm, or how fast a hero can farm, which are both interesting questions.

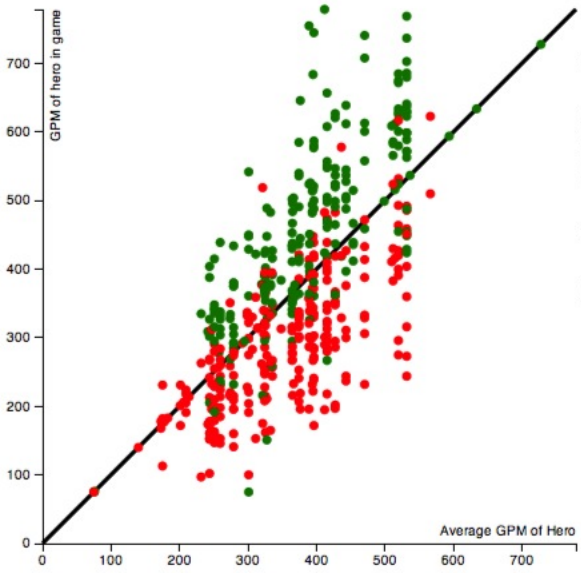
Future Development

1. A good question this graph could answer is "what happened in that game???" if the game is "worse" than average. We need to have an on click event, that brings up the end-game screen of the respective game.

Figure 14: Design Process

Designing the Statistical Graphs- the GPM/XPM Graphs IV, April 5 Update

Angela Fan



Looks pretty good.

Heroes are lined up in lines based on the x-axis, which are heroes that are played often.

Has a tooltip mouseover.

Need to have on-click event.

Figure 15: Design Process

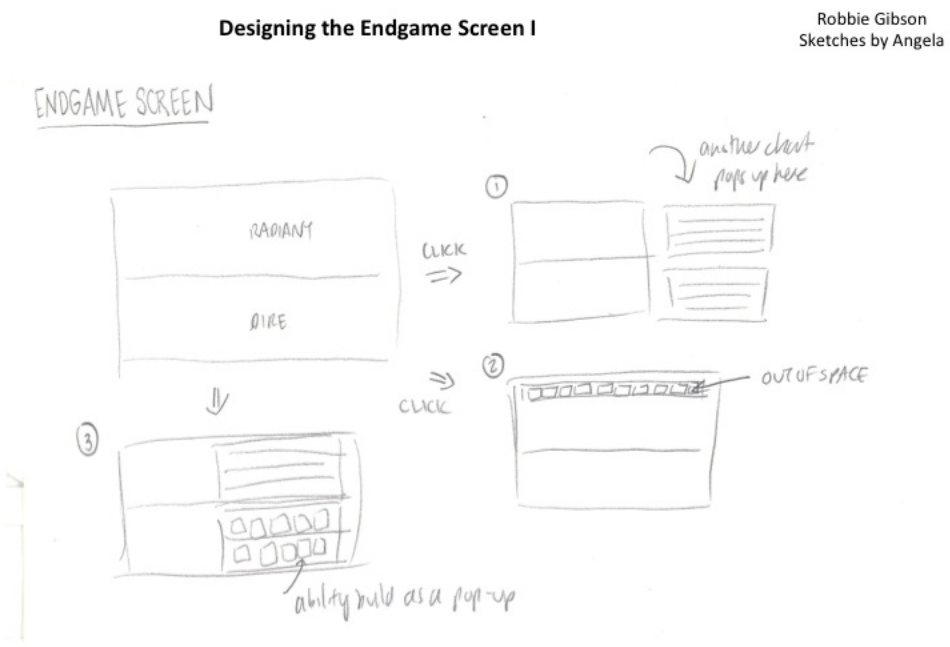


Figure 16: Design Process

Figure 17: Design Process

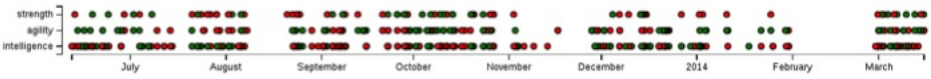
Designing the Timeline Benjy Levin

Design Process and Evolution:

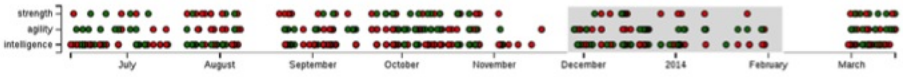
1. We originally planned to have a timeline down the left side of the page, displaying each user's games. Has a brush feature so users can zoom in to a certain time period.
2. Decided to implement this using the image of the hero that the player played in a given game. However, upon implementing this solution, we realized that a given user could typically have 500+ games and thus 500+ icons, which were far to many to scroll through in a div with scrollbars.
3. In addition to this issue, we realized that time is more intuitively perceived linearly from left to right, not up to down or down to up, and when a brushable timeline is implemented, it is far better design practice to have the entire time period in view, rather than having to scroll.
4. Decided to pursue a different implementation.

Figure 18: Design Process

Designing the Timeline Benjy Levin



1. Overview of all games played over time, divided into three rows based on the main attribute (strength, agility, intelligence) of the hero you played in that game. Colour encoded as red for win or green for loss for that game.



2. Timeline is brushable (see highlighted section above), and upon brush, the graph will zoom into that area and redraw the x axis with more detail based on this new selection – see

3. Upon zooming in, a "Clear Brush" button will appear in the bottom right, that will logically clear the zoomed in brush when clicked.

Figure 19: Design Process