UNIVERSITY OF BRITISH COLUMBIA, DEPARTMENT OF COMPUTER SCIENCE

Project Proposal

Ben Janzen, Giovanni Viviani
bjanzen@cs.ubc.ca, vivianig@cs.ubc.ca

October 31, 2014

1 Domain

DOTA (Defence of the Ancients) 2 is a fast paced, free to play action game based upon a custom map from warcraft 3. Since getting a stand alone release DOTA 2 has become one of the most popular online games in the world, clocking in over 600 thousand unique players online in a single day (stat taken Tuesday 28th October 2014). In this game two teams of 5 players compete. Each team has an ancient (a building) that they have to defend from the other team. The ancient is defended by a series of stationary automated guard towers. Both teams also have armies of non-playable soldiers that slowly stream towards and attempt to attack the enemy base. Each player chooses a hero to play out of a pool of over a hundred (usually with limited knowledge of the enemy team's selections). As a single match progresses players earn experience to spend on 3 basic abilities, and eventually one ultimate ability (all of which are unique to each hero). By killing the creeps or buildings of the enemy team players are able to earn money (called gold) that they can use to purchase items that improve their heroes.

As is to be expected in such a complex game, the strategic trade-off space becomes extremely large and unmanageable. The game is under constant development and balance tuning. The recent 6.82 Rekindling Souls update is perhaps the most extreme example of this. The map layout was changed significantly, two heroes were remade almost entirely and almost every hero or item had at least some minor tweak. With matches potentially lasting a full hour, it is simply impossible for any one player to play the hundreds of games required to test out all possible item-skill builds for all possible heroes and see for themselves which strategies are powerful. Thus players tend to turn to online tools and guides in order to learn about the meta game. Meta game is a term used within the DOTA 2 community as a term to denote what heroes are considered strong, how often they are played and how they are built (what items and skills a player should purchase and in what order).

2 Dataset

Valve (the company who owns and maintains DOTA 2) provides a public web API that exposes detailed statistics about every match played (by users that have consented to having their match history shared through a checkbox in the options menu). According to Valve, approximately 50,000 games are recorded per day, and records are maintained for one month. This data includes (but is not limited to)

- Skill level of players
- Which heroes were played
- What items they had at the end of the game

- What skills did each player purchase (and when)
- Which team won
- Match duration
- Aggregate statistics about hero performance (gold/experience earned, damage to heroes and objectives, creeps killed, enemy heroes killed ect)

3 Task

We abstract the task of meta game analysis and learning into the more generic terms of trend analysis and outlier identification. Most lay users simply want to figure out which hero they should play. We abstract this query as a simple outlier identification task. Specifically player's looking to win as much as possible want to find overpowered (too strong) strategies and players looking for a challenge want to find underpowered (too weak) strategies. However, DOTA 2 has a strong emphasis on competitive play, and strategies are rigorously discussed and debated. Many players want to inform their arguments about balance and strategy with knowledge of how trends in strategic choices change over time. For example, one may ask whether just one or two strategies are constantly popular, or if their is a cyclical relationship going on (A beats B beats C beats D beats A). How do these trends change on a Saturday night versus monday morning? Do major tournaments and patches have an effect? We will aim to provide a framework for a broad set of trend analysis tasks over a dynamic set of time series data. It's also important to note that these analyses are not necessarily limited to pick rates. We also want to incorporate win rates and intelligent metrics of hero/skill/item performance based upon their strategic classifications.

4 Personal Expertise

All project members are enthusiastic DOTA 2 players with several hundred games played each. To a greater or lesser extent, we follow the meta game from professional e-sports games and participate in frequent LAN parties (events where a group of friends get together to play DOTA over a local area network). We are looking to create tools that help improve our understanding of the meta game, and hopefully these tools will be helpful to other players.

5 Previous Work

The classic method for supporting meta game analysis is the tier list. These are tables (distributed on reddit, or the official forums), usually compiled once a month showing pick, ban and win rates of heroes in top end competitive play. They are usually accompanied by several pages of explanation by a professional player, and heroes are binned into a few categories based on how powerful they are. Players looking to choose an overpowered hero will select tier 1 heroes whereas players looking to choose underpowered heroes select lower tier heroes. This solution is of course incredibly aggregated and is only able to detect the most broad of trends. It is also very difficult to get any sense of change over time without reading dozens of pages of these threads.

There are a number of websites already taking advantage of the match history API. Most notably DOTABUFF is designed to provide the raw statistics of hero, player and item performance over time. However, most of these applications do not curate their data intelligently, and provide only the most basic of visualizations. For example, each hero has a view showing a bar chart of the top X most bought items with bars showing the win percentage of each. However, since the item list is a snapshot at the end of the game, these stats are highly misleading. It makes logical sense that if a player has an early game item in their inventory at the end of the game, they will have a low win rate. If they were going to win, they'd probably make enough gold to sell it and buy a more expensive item, but they may have lost even worse if they didn't buy it to begin with. We aim to provide more intelligent visualizations and support more accurate analysis than these other sites.

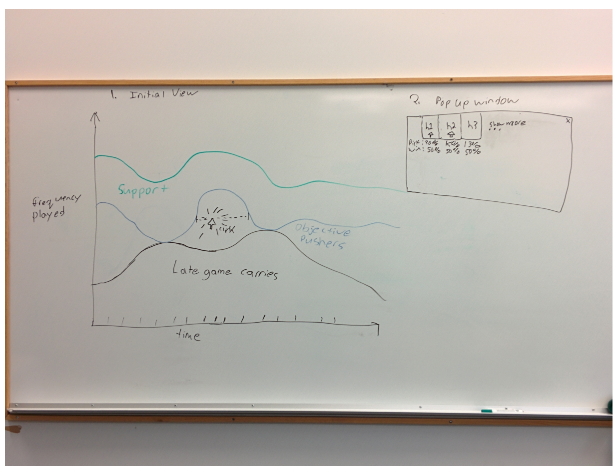
The game provides some rough initial work classifying the different heroes in DOTA 2 based on roles they can perform, but this is far from exhaustive, not always accurate and can definitely be expanded upon. Any classification of finer grain strategic choices like items and skills themselves are not present, and will need to be created from scratch.

Fundamentally our problem is one of representing time series data with appropriate aggregation and attribute reduction. These are well studied topics in visualization literature. Our particular task is focused almost entirely on temporal trend analysis of potentially massive datasets (of a limited number of attributes).The Steam graph is particularly effective for these tasks, and crucially, has been argued to be effective even in the hands of the lay public [1]. While the project contributors are the primary stakeholders in this visualization, ideally it will be consumed by a broad range of DOTA 2 players. Thus having a simple and easily understood encoding, such as a stream graph is highly important.

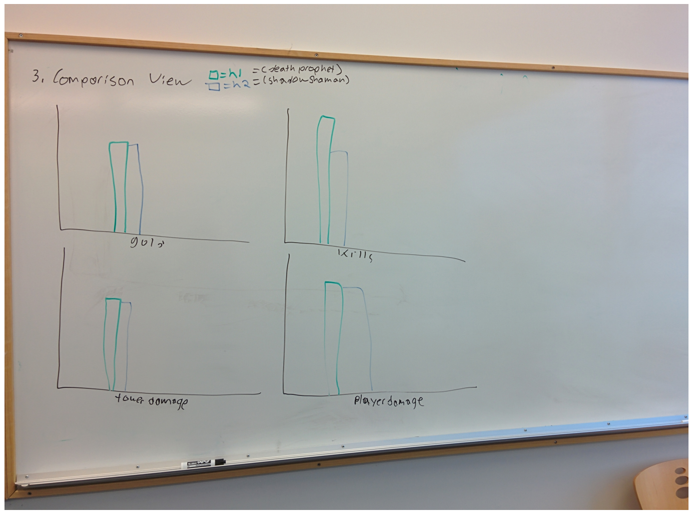
6 Our Solution

Our initial design discussions have focused around using a stream graph support trend analysis by visualizing changes in hero, item and skill choices over time (for conciseness, only the hero use case is described below). Before beginning our implementation we need to leverage our expertise in the game to create a basic taxonomy of items, skills and heroes. We've identified that one of the main failings of previous solutions is not properly framing the raw statistics and aggregated results in a useful way. In particular there are frequent issues with comparing across groups of strategic choices. For example, comparing gold earned between Treant Protector (a hero designed to assist friendly heroes) and Riki (a hero designed to assassinate enemy heroes and farm) is not valid. Ideally this taxonomy will have a small enough set of bins to be visualized in a nice stream graph that allows us to understand trends in bins over time. Interaction through mouse selection and highlighting will be used to drill down into subsets of time series data, drill down into more detail about particular bins, and compare results in juxtaposed views.

7 Example use case



1. A user enters our system, and is presented with a stream graph visualizing hero pick rates (aggregated into our taxonomic bins) over time. The user spends a great deal of time analyzing these changes looking for trends of interest and outliers.
2. As the mouse hovers over a particularly large objective pusher section a pop up dialog will appear showing the top X most picked pushing heroes
3. Surprisingly the most picked hero, Death Prophet is picked twice as much as the next picked hero, Shadow Shaman but their win rates are approximately the same. The user select both heroes for a more detailed comparison.
4. In the side by side comparison it is revealed that Death Prophet gets more gold and kills more enemy heroes, but damage dealt to towers and overall win rates are similar.



5. The user concludes that while both heroes have their place and seem to be valid picks, Death Prophet is picked more often due being able to fill a more general role and decides to play a few games as Death Prophet.

8 Implementation

We will be creating an automated server to poll the match history API every X minutes, and populate a local database (as it would be to slow and inefficient to continually poll the API on demand). This database will be managed using Django[1] to provide a database independent schema. Django also provides a basic web server (that can be switched out later for production deployment) that should work perfectly to serve our planned D3.js visualization.

9 Milestones

1. Script to read API and drop data into local database Nov 14
2. Data cleanup Nov 14
3. Create Taxonomy of skills, items and heroes Nov 14
4. Literature Review, focusing on temporal data encoding Nov 14
5. Stream graph prototype Nov 28
6. Add interaction and pop ups Nov 28
7. Scatterplot matrix view Nov 28
8. Testing and design iteration on going till end of term
9. Paper write up on going till end of term.
10. Adding additional views if possible

References

[1] Lee Byron and Martin Wattenberg. Stacked graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, November 2008.

[1] `https://www.djangoproject.com/`