Interactive Visualization Final Project Report: Dota 2 Simple Replay Viewer

Brandon Ip, Nathaniel Wheeler

April 26, 2018

Introduction

Our visualization project is centered around an online game. We'll discuss more details about the different mechanics of the game, but here is a quick intro. Dota 2, short for Defense of the Ancients, is a MOBA game (Massive Online Battle Arena), where matches are played 5 vs 5, with each player controlling a unique hero with unique abilities. Each team controls a base, which includes a structure known as the Ancient, and the objective of each team is to destroy the other team's Ancient.

Our initial project idea was to visualize the movement of the players and/or the creeps (AI controlled NPCs) throughout a match. Along the way, based on peer feedback and our own experience, we discovered that a simple replay viewer would be an even better goal. Movement is only a small subset of data available in the replays. Why not visualize the rest?

Motivation and Purpose

Our initial motivation was a bit specific. Players like ourselves who want to improve at the game would find that a visualization of player and creep movement throughout a game is easy to analyze. But why stop there? Including other useful information, including kills, deaths, gold, and status of each player, provides more room for analysis. However, we don't want to flood users with information, else that would contradict our goal of easy analysis. So we've decided to include only useful data such as the previously stated examples.

Design

The tool is divided into three sections. On the left is the map, on the top right is the scoreboard, and on the bottom right are the options. Here, we'll discuss details about the game and how they are presented (or will be presented in future versions) in the visualization.

Gameplay occurs on a single map that is used for each match. The layout changes slightly as Valve (the developer company of the game) releases updates, but the general layout is the same. Each of the two teams (named Radiant and Dire) control a base, and these are located at the bottom left and top right corners of the map. The map features varying terrain, including three lanes, one going along the top left corner, another going along the bottom right corner, and one straight across from bottom left to top right. In our tool, we use the map as a background image and overlay data over it. Since the layout changes slightly across game versions, we'll need to have multiple map images to apply the appropriate one for the replay. Due to time constraints, this is not implemented yet. The current image used is for patch 6.82.

Since we are dealing with 3-dimensional timed data, there needs to be control over time. A slider at the top of the page lets users control which time frame of data is visualized. In case the user wants to be more precise, the time can also be entered in the text fields to the right of the slider. In case the user doesn't want to control time themselves and instead wants to view the replay in real time, there is also a play/pause button to the right of that which animates the visualization by stepping forward in time every set interval. The interval, which controls the play speed, can be controlled with a slider and text field in the options section.

A circle is used to represent each player's position at a certain time frame. That means there are 10 circles for most of the game (in earlier patches, not all players spawned at once in the beginning, rather, players spawned when they chose their hero and entered the game). Each circle is assigned a color from a color scheme (taken from ColorBrewer) to differentiate each player. The circles are bordered green to indicate the Radiant team, and red to indicate the Dire team. When a hero reaches 0 HP and dies,

the players are instead represented by squares. We considered representing the players with the hero sprite icons. Based on feedback from peers, we found that it was not that important. Most people thought that colored circles would be easier to differentiate. We will add the option to toggle between circles and icons in the future.

There are AI controlled NPCs (Non-Player Character) that spawn during gameplay, called creeps. There are two types of them, neutral and lane creeps. Neutrals will spawn in areas of the map known as the jungles, and lane creeps spawn at each team's base and try to make their way to the other team's base through the three lanes. In addition to the players, we had planned to visualize the movement of creeps also, but could not due to time constraints. However, we do have the data parsing complete, so it should be implemented soon.

To visualize movement over a course of time, a trail is drawn over the previous positions of each player (and creeps, once their visualization is implemented). They will match the color of the circle and be smaller in radius to reduce clutter and differentiate from the actual current positions. The trail length is defaulted to 5, but can be controlled by the user with a slider and text field in the options section. The possible values range from 0 to the entire length of the game.

The scoreboard provides some of the player stats at the selected time frame. These include hero kills, deaths, creep kills, and gold. Players earn gold by killing neutral or enemy creeps or enemy heroes, and they can spend them on items to further increase their advantage. We considered providing this info instead as a tooltip that shows when hovering over the circles, but that makes it difficult to compare the stats across players. In the future, we will add more important stats to the scoreboard such as HP/MP, items, and possibly skill cooldowns. Also on the scoreboard are the player icons shown on the map. With this, users can match up the player data on the scoreboard with the player position on the map. Clicking the icons on the scoreboard toggles drawing of the dot and trail on the map of that player.

There are various structures for each team, including the ancients, the towers which are stationed in each of the three lanes and which attack any enemies, and

barracks which, if destroyed, strengthens the other team's creeps. The status of these structures are important to visualize, but are not currently implemented yet. Since they are stationary and are in the same positions in every game, a single shape and color should be sufficient to represent each structure. Users who know the game would be able to differentiate them based on their position.

Prior Work

The replay viewer built into Dota 2 is comprehensive, including everything possible using the actual interface and models used in the game. This may be troubling for viewers who don't have or don't want to use the Dota 2 client, and for viewers who believe that it provides too much information, making it difficult to analyze gameplay. The tool we have created acts as an alternative, and aims to simplify the replay data by showing only the most useful information in a compact view in a web browser, which is easily accessible by anyone.

Dotabuff is an online website community managed and developed by a small team dedicated to gameplay analysis. Unless a user is subscribed, Dotabuff only provides the stats at the end of a match, and only certain stats throughout a match. Subscribed members will have access to an extremely wide variety of visualizations of statistics throughout matches. However, the puzzling thing is that they don't have a visualization of player/creep movement throughout a match. This was a very important motivation for our project.

Design Evolution

The core design of our project remained mostly unchanged through development. Our original plan included visualizing the movement of players and other entities throughout the course of the game, filtering these entities to reduce clutter, having a scrubber to allow the user to move through different moments in the game, as well as a play button to let the game run in it's natural progression. The first hurdle we overcame in tackling these goals was finding a way to take the information embedded in

a replay file and export it into usable data. After reviewing many candidates, we settled on clarity for two reasons: it works with both the new and old versions of replay files, and it is written Java, which both of us are familiar with, making it easier to extend than the other candidates. Using clarity, we managed to extract position data from replay files as well as other relevant information with a few simple calls. After collecting the data, we needed to decide on a platform for the visualization itself. We elected to use d3 because of its elegant approach to handling with large data. Our initial/test design was simply a scatterplot of every single datapoint recorded at every point in time. This quickly evolved into the version we used to gather feedback from peers, which separated points by timestamp and had a slider to switch between frames. Based on that feedback, we prioritized the features we implemented next. One of the most popular requests was adding a way for users to easily distinguish which player was on what team. This was the next feature we implemented because we thought it was essential to understanding the flow of the game. Another feature we added was a way to tell if a player was dead or alive; we used square icons for dead, and circle icons for alive. Another QoL change we made was to the legend: while players' icons could be easily tracked, it was hard to find which hero each icon represented. To combat this, the legend was sorted by team, which seriously alleviated this problem. The next two iterations of added player trails and player filtering, to help users identify individual players and their trends better. The final update as of writing this paper added a scoreboard to track player stats as well as a play button with a speed slider to allow the game to progress automatically.

Feedback

Much of the feedback from our peers included wanting info such as kills, deaths, and HP of players and location of towers to be available. Thanks to this, we expanded our vision of our project from visualizing just player movement to visualizing other game data. Unfortunately, for such a large project, we weren't able to implement a lot of the planned features due to the short amount of time given.

Here are other peer-suggested features that we did implement:

- Autoplay: Animate the visualization, controlled by a play/pause button and a slider for play speed
- Events (dying): Indicated by circles changing into squares
- Scoreboard: A table of players stats updated throughout the match
- Team colors: Indicated by the color of the dot borders
- Filtering: Allow user control over which players to show on the map
- More precise input: Text fields next to sliders
- Player paths: Indicated by the trails following the dots on the map

Conclusion and Future Work

While we believe our visualization in its current form would be an excellent tool for serious Dota players, there are some features we have not yet implemented that would truly make app robust. The biggest missing feature from a technical perspective is the ability for users to upload their own replay file; right now, the data is pre-parsed and is loaded by d3 when the webpage loads. The other technical issue we have is while clarity supports both versions of replays, our program only uses the older version (we're hoping to fix this one pretty soon). One feature that is currently in the works is tracking creep positions and stats. We have implemented a way to grab the data, but have not yet utilized it in our program. Along those lines, we are interested in collecting and displaying stats related to towers and other structures to give users a better idea of who has the map advantage. We would also like to have health bars displayed on the various entities on the map. One idea would be to have the outline of the each character to represent the percent health that hero has, with an option to mouse over the character to get more details. Two more minor planned features would be to implement a way to view characters' items and toggle the icons between circles and sprites.

Division of Labor

- Approximately 40 hours spent on this project

- Brandon Ip:
    - Parsing player data
    - All UI and features
- Nathaniel Wheeler
    - Parsing creep data
    - Setting up d3 and the initial visualization

References

Papers:

A. M. MacEachren, F. P. Boscoe, D. Haug and L. W. Pickle, "Geographic visualization: designing manipulable maps for exploring temporally varying georeferenced statistics," Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258), Research Triangle, CA, 1998, pp. 87-94, 156.
doi: 10.1109/INFVIS.1998.729563

S. Kriglstein, G Wallner, M. Pohl, “A user study of different gameplay visualizations”, ACM DL, CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems
Pages 361-370

Game Resources

https://www.dotabuff.com/

https://github.com/skadistats/clarity-examples

http://colorbrewer2.org/