A Visualization of DOTA2

Fredrik Johnson

Abstract— This report will explicate different ways to visualize a game of Dota2. By visualizing data aquired in an easy to use view, interaction should prove easy. The result of this product hope to help players understand their impact in the game and how they can improve in their specific role. By letting players and teams enter their own played games it provides tools for analyzing and excelling their game play and take their skill to the next level

Index Terms—Multivariate data, Temporal data, Sports analysis, Computer games

1 Introduction

Competitive gaming (Esports) has been on the rise since the early 2000s and is on its way to become a socially accepted occupation with its high price pool tournaments and streaming revenues. One of the most popular games in Esports is Dota2, a fast paced, team based game where two teams face each other in a battleground with the goal of taking down the enemies main building, the ancient. The few tools that already exists chooses to show a players life time statistic in tables as numbers. While this report will instead focus on presenting and visualizing vital information from only a single match. Having spatio temporal data and numeric values this report will try to evaluate the best way of visualizing these.

Disclaimer: Without prior knowledge of Dota2 or similar games this report might not be understood to its fullest degree

2 Background and Related Work

Since Dota2 and Esports as a whole is relatively new there are not as much work done as you would find in other sports such as hockey, baseball etc. There are some tools that tries to visualize data acquired from Dota2, however these are either focused on just presenting raw data or animations of movement in a match; where the animations are often done offline and uploaded as a movie or GIF.

The tool that is currently established in the world of Dota2 is Dotabuff[1]. Dotabuff presents a lot of life time data and statistics such as win rate with a specific hero, item trends etc. These are presented in tables and graphs, which can be nice for seeing change over a long time span. Devilesk[2] displays an offline rendered animation of the movement in a match which is primarly done for artistic purposes and not for analyzing. Both of these tools are nice in their own way however none of them gives a quick overview of the state of the game.

In contrary to the previously mentioned tools, this visualization tries to combine both of them and presents the user with an easier way of telling how a match ended. While at the same time give an overview of how the teams utilized the map.

3 Data

The data used in the visualization was obtained through parsing demo files(.dem) of played Dota2 matches. These demo files contains everything that happened during a match. however they are compressed using protocol buffers, this makes the extraction of the data harder. There exists a few very undocumented libraries(Alice[3], Smoke[4], Clarity[5]) which provides an interface to help out with the extraction of usable data. This visualization used clarity, a java based implementation of the original python based Smoke.

There was two different data sets generated from the demo files. This reports first focus was obtaining the positional data, this data set contained every players position for about every second. Since the demo files record changes at each server time step some seconds might not contain new information of position for a specific second. In an average game which last for about 40 minutes, there would be 2.000 entries for each player.

Using the obtained positional data another value was calculated, the distance a team kept from each other. This was calculated by calculating the mean position for all players(centroid), thereafter removing the player furthest away from the given centroid and recalculate the centroids position. The player furthest away was removed because that would lower the spikes caused by a player moving a far distance during a low amount of time.

The second data set was a much smaller set, it contained the statistics at the end of a match such as enemy players killed, amount of deaths etc. This data set was much easier to keep track of since it would always be a fixed size.

All the data was written and saved into formatted CSV-files which the web application can read.

4 Method

The tool was divided into four different parts which all uses one of two different data sets. The first set containing the positional data was plotted on a overview version of the map which is an overhead view of the in game map. Each position is illustrated by a circle which opacity depends on how many circles are currently shown. The color of the circle depends on which mode which is used, there are two different coloring schemes. The first is a temporal color scheme which varies between blue-yellow-red representing the early-, mid- late game. The other is a qualitative color scheme where each player has a designated color which is used for all the different visualizations. Both modes are present because they both has it uses, the temporal mode is great for getting a overview of how the match played out; while the qualitative mode is necessary when there is to see where each specific player where during e.g a team fight.

There was some discussion if the temporal scheme should had used a two-colored schemed as proposed by [2], however when using a two-colored scheme the middle of the map will show up as gray. This was not an acceptable scheme because it was too hard to distinguish the transitions between the different stages of the game.

To interact with the map the time axis below the map is available for brushing, an arbitrary interval can be chosen to be shown at any time of the game; it is also possible to scrub through the game with the chosen interval.

The second visualisation is a proximity graph which uses derived values from the positional data, this graph shows how close the team has stuck together throughout the game. As explained before the graph only takes the four closest players into consideration, this is because players can teleport over the map which causes huge spikes in the graph. Also a player can be at another part of the map but still be helping out the team and if those players where considered into the

• *Fredrik Johnson, Linköping University, Sweden, e-mail: frejo989@student.liu.se.*

[1] http://http://www.dotabuff.com/
[2] http://devilesk.com/dota2/apps/replay/viewer/
[3] http://github.com/AliceStats/Alice
[4] http://github.com/skadistats/smoke
[5] http://github.com/skadistats/clarity

calculation the graph would be misleading. The proximity graph is connected to the time axis the map uses so it is possible to select and arbitrarily interval in the graph and the time axis will adjust after the selected area.

A table is used to present the end game statistics next to a players hero, color and name. The values presented in the table are specifically chosen, because they are values important to specify the role and the impact to the game done by the player. To help determine the max values in a table with a lot of numbers a heat map coloring is used which range from a deep to light red hue, where deep red represent the highest value and the light the lowest. The table also accepts user input in the form of clicks, a user can select a specific team or player and this will update the map to only show the corresponding movement patterns.

The last visualization is a radar chart, tightly connected to the table. Even though the table uses a heat map it can still be messy to use it, a radar chart gives the user an easier way to look at values more frivolously; since its easier to interpret shapes than raw numbers. The shapes seen in the radar graph quickly indicates what role a specific player had in his team.

At first the radar chart will show the average values, derived from the end game statistics, using the same colors at the proximity graph. This is done with the purpose of avoiding clutter and to give a fast overview of the match. When a player is selected in the table the radar chart will update to show the chosen player compared with that players team average.

5 Implementation

Since the solution was supposed to be easily accessible the visualization was implemented as a web application where primarily two JavaScript libraries where used, D3.js and jQuery. D3.js is a library which allows manipulation of DOM elements using standard web elements while jQuery was used to simplify events happening when using the product. These two specific libraries where chosen so that every one with a standard web browser could utilize the product without the need of installing anything external.

There is a notable difference between broswers using the product, which is probably caused by the JavaScript engine implemented for that browser. Since handling DOM elements is a slow process and there is a lot of data, about 20.000 SVG elements, some browsers did not have a fast response when brushing the time line.

6 Results

The result is visualization which combines a new way of looking at the data and old elements that players are used to see.

There is a map which shows the position over time, instead of having it animated it shows the whole movement pattern immediately, *fig 1*. It is also possible to choose another color scheme that will instead of time dependent show the player color for each circle.

Since team play is a vital part of Dota2 there is a proximity graph, *fig 2*, below the map, The proximity graph uses the same time line as the map to make it easier to use the two visualizations together. In the picture an interval has been chosen to show the scrubbing technique.

The table, *fig 3*, shows the statistics at the end of the match, a table is a recognized way of viewing data for any player. To make it even easier to understand a heat map is used to colorize the values from a deep to light red color, where a deep red represent the highest value of that category.

A radar graph, *fig 4*, which shows the same statistic that the table uses but in an way easier to interpret.

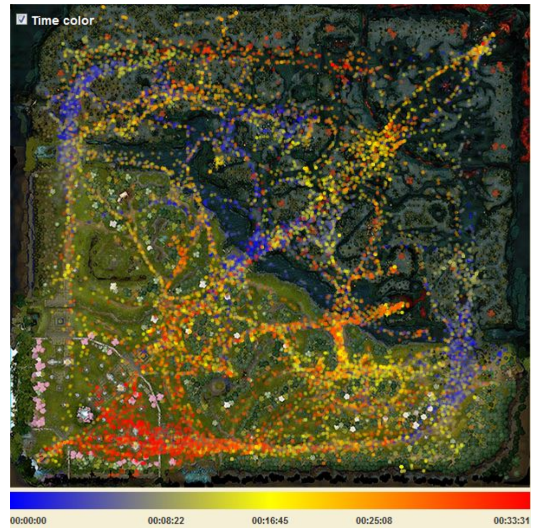

Fig. 1. Map showing players movement patterns

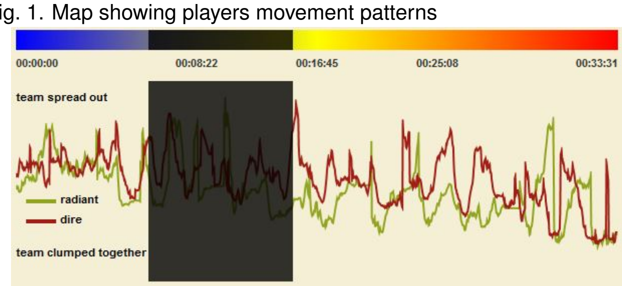

Fig. 2. Proximity graph as a function of time

Name	Levels	Kills	Deaths	Assists	Gold	XP	LH	Denies
Show all								
Show Radiant								
fy	14	2	9	8	10572	11470	88	0
Super	18	5	6	8	17328	17550	219	14
Fenrir	11	4	6	5	7285	7066	18	0
iceiceice	11	2	6	4	9019	8184	73	8
Black^	14	2	5	4	13909	11661	166	18
Show Dire								
ppd	15	2	2	15	9226	11955	61	2
SWAGKID!	21	17	7	11	21340	23894	232	5
UNiVeRsE	16	3	3	23	11248	14441	100	4
Aui_2000	17	4	2	16	15401	16492	172	8
Fear	20	4	2	6	20257	22300	260	8

Fig. 3. Table presenting the end game stats

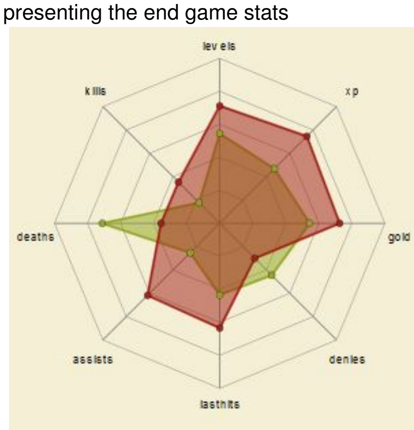

Fig. 4. Radar showing the teams average

7 EVALUATION

The target audience for the product is Dota2 players. The evaluation done for this product was influenced by methods described in [1].

Since our product has quite the small audience of choice, participants for the evaluation where chosen specifically with consideration of previous experience with the game. The test was carried out the same for all participants with exception for the last question which was conducted as a two variants test.

First the participant was asked to tell which team won the game, using just the map with the temporal data. All participants succeeded with the task of telling the winning team, where the time of the answer spanned between 5-10 seconds.

To evaluate the time scale and color mapping the participants where asked to select to only select and show a small portion of the early game. Thereafter scrub through the time to the late game. Almost every participant completed the task without further difficulties while they also explained what the color representing the time of the game meant. How the colors corresponded to the in game time.

As mentioned earlier the final part of the test was carried out as a two variants test, where half of the participants used a table featuring a heat map coloring while the other half used a table without the heat map. The participants where then asked to point out which player had the most impact on the game in both teams and what specific role each player had. Players familiar with the game should have no problem extracting this kind of information from the table and numeric values alone. So the test wanted to see the response time for each participant if the heat map helped out or not. Participants whom used the table with a heat map had a response time of 2-3 times quicker than participants without the heat map. This result was considered a success.

8 CONCLUSIONS AND FUTURE WORK

There is still room for a lot of improvement with this tool, at the moment it mostly shows gathered data in a fancy way. It is hard to get any meaningful insight from the current iteration. However since a lot of time was spent trying to acquire the data and not focusing on the visualization, it's still something that can be used for a fast and accurate overview of a match.

A vast improvement would be to let the user upload their demo files and the tool would parse them and then visualize them, at the moment the tool only accepts formatted CSV-files. The user would need to use an external application to parse the demo files and then upload them to be used in the visualization. There are frameworks that can convert the written Java application into javascript such as GWT[6] so it can be runned entirly on the web. However we did not have any time too look into this during the project.

REFERENCES

[1] C. Forsell and M. Cooper. A guide to reporting scientific evaluation in visualization. *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 608–612, 2009.

[2] K. Moreland. Diverging color maps for scientific visualization (expanded). *Proceedings of the 5th International Symposium on Visual Computing*, 2009.

[6] http://http://www.gwtproject.org/overview.html