

Lab 5.2 Exploring Spatio-Temporal Player Behavior in DotA 2

Load up StratMapper

Please go to the following link: <https://sm-dota-v2-ro.herokuapp.com/>.

Data

The dataset used in this lab comes from the game *Defense of the Ancients 2* (DotA 2). The data comes from two matches from a tournament league. It includes play-traces for all twenty players (ten in each game, five on each team). A player's trace includes, among other details, the hero they are playing, kills, deaths, and location over time. StratMapper displays this information visually by overlaying it on top of a timeline and a game map.

The raw data is arranged per match in a csv file format. The data records several types of events. Each line in the csv file is an event with a match id to relate the event back to the match. The table below shows the list of these events and the corresponding contexts. Each recorded event has a timestamp identifying when this event has happened.

Event Name in Raw Data	Event Context
DOTA_COMBATLOG_DAMAGE	Someone is damaged
DOTA_COMBATLOG_HEAL	Someone is healed
DOTA_COMBATLOG_MODIFIER_ADD	A buff/debuff is applied
DOTA_COMBATLOG_MODIFIER_REMOVE	A buff/debuff is removed
DOTA_COMBATLOG_DEATH	An entity dies
DOTA_COMBATLOG_ABILITY	An ability is used
DOTA_COMBATLOG_ITEM	An item is added to a user's inventory

DOTA_COMBATLOG_GOLD	A player experiences a change in gold
DOTA_COMBATLOG_XP	A player experiences a change in XP
DOTA_COMBATLOG_PURCHASE	A player makes a purchase
DOTA_STATUS_UPDATE	This is an "OnTick" event that is artificially created to get status updates from players periodically throughout the game

Each event has different types of attributes. For example, *DOTA_COMBATLOG_DAMAGE* event is recorded when someone is damaged in the match, and thus will contain the following attributes:

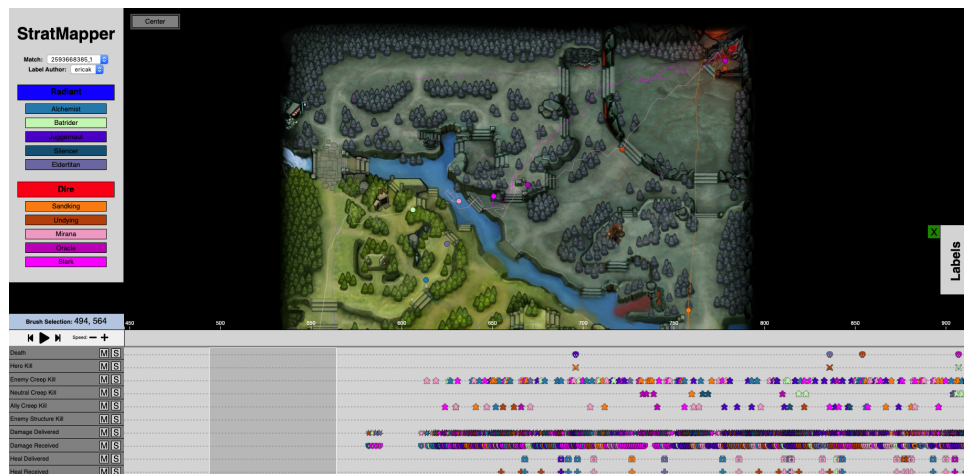
- *timestamp* identifying when the event occurred,
- *source_name* the name of the entity who caused the damage,
- *target_name* identifying who was damaged,
- *inflictor_name* is the name of the object that caused the damage,
- *attack_damage* is the amount of damage caused, and
- *health_remaining* which identifies how much health is remaining for the entity who received the damage.

Another event worth noting is *DOTA_STATUS_UPDATE*, which is an artificially created event to get the status of each player periodically throughout the game. This status information includes player position, total gold, total XP, item list, and some other important attributes that can be used to understand the player status throughout the game. Using this event, we can track all player information over time during matches.

As you learned in previous chapters, we pre-process this data and create a cleaned and processed data to be visualized by StratMapper. We will not go through the details of the data preprocessing and abstraction. However, we will use the data we preprocessed, which is available in the folder for the lab, so feel free to inspect it before you begin using the visualization system. Note that the data is in JSON format, which is the format needed for StratMapper. If you have problems understanding this format or the data, consult the many tutorials on the web about the format.

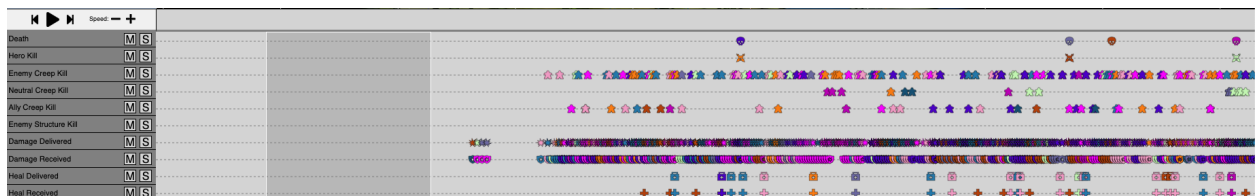
Interacting with StratMapper

When you open *StratMapper* from the link provided you should see the following:



The upper part of the screen is the game map, the lower part is the timeline. You can zoom in and out on both using your computer's scroll feature.

Take a look at the timeline, it is filled with colorful icons that represent players' actions. The location of an icon on the timeline represents when during the match that action occurred. Click on the gray highlight and drag it across the icons. If you watch the map while doing this, you will see the trails of each player's move through the game space. Action icons will appear and disappear on the map as you track over the corresponding icons on the timeline.



You can also **play** the data using the play button:



Pressing the play button will play back the game given the time window highlighted. You can also change the width of the highlight by clicking and dragging its edges. Try doing this and see how it affects what is displayed.

Heroes can be highlighted or muted by selecting their names in the sidebar (see image below). Similarly, actions can be shown or muted by selecting **m**, for muting, or **s**, for shown, next to their names in the timeline sidebar. Try muting and unmuting different

heroes or actions as you move the grey highlight around to see the effects of such actions.

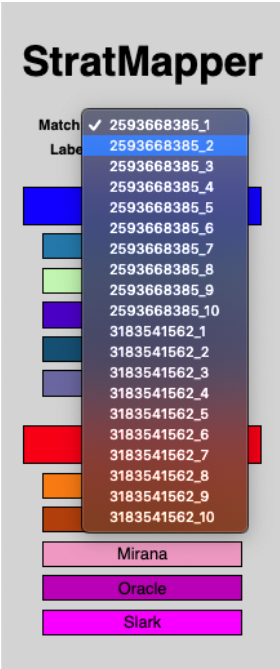
Radiant	Death	M	S
Alchemist	Hero Kill	M	S
Batrider	Enemy Creep Kill	M	S
Juggernaut	Neutral Creep Kill	M	S
Silencer	Ally Creep Kill	M	S
Eldertitan	Enemy Structure Kill	M	S
Dire	Damage Delivered	M	S
Sandking	Damage Received	M	S
Undying	Heal Delivered	M	S
Mirana	Heal Received	M	S
Oracle			
Slark			

For further details regarding *StratMapper*, please refer to chapter 5.

For now, you can take this time to explore *StratMapper* functionality. Use the filters and the playback to explore the data. What kinds of behaviors do you see? What conclusions can you draw about the players' strategies?

Labeling with *StratMapper*

Once you've identified a behavior that you think is of interest, you may want to mark it, this is where labeling comes in. Use the dropdown in the sidebar (see image below) to go to a match that is already labeled.



You'll notice that there are now small highlights with text in them in the bar above the timeline:



These are behavioral labels. Normally, *StratMapper* allows one to create their own labels and place them, but for the purposes of this lab, you will not be creating your own labels and will instead be interacting with labels that were derived and placed by the research team. For information on how to write, edit, and delete labels, please refer to Chapter 5.

The labels used in this lab are as follows:

Label	Full Name	Description
TF-OS	Team Fight - Objective Struggle	The team fights over an objective such as a tower.
TF-R	Team Fight - Retaliation	The team fights back to get revenge for something the other team did
TF-FT	Team Fight - Focus Target	The team fights in order to take down a particular player

SR-F	Solo Recovery - Farm	The player dies and then kills non-player entities alone after reviving
SR-S	Solo Recovery - Scout	The player dies and then roams the map alone after reviving
SR-P	Solo Recovery - Push	The player dies and then moves the center of action closer to the enemy side of a lane (alone) after reviving
TR-P	Team Recovery - Push	The player joins others to move the center of action to the enemy side of a lane after reviving
TR-OS	Team Recovery - Objective Struggle	The player joins others to go after an objective, such as a tower, after reviving
TR-A	Team Recovery - Assist	The player helps another player after reviving

With the recovery labels dealing with the actions a player took after dying and then reviving back at base.

Pick a label of interest and click on it to see what happens.

If you look at the sidebar where the hero names are listed, you'll notice that some of them are now muted, and those that remain highlighted are the heroes who were involved in this specific labeled behavior, as indicated when the label was placed. Click on the label again to un-highlight it.

Analyzing Labels in StratMapper

Using the same drop-down menu as before, move between selections two and nine and examine the labels that have been placed.

Things to think about:

- Who is involved in the label?
- What events are encompassed by the label?
- Where on the map did the labeled event take place?
- When in gameplay did the labeled event take place?
- What happened before and after the labeled event?
- How often does this label appear?

Using the provided labels, and these six points, try to identify at least one behavioral or strategic pattern.

Things to think about:

- What is the pattern or strategy?
- How was it identified from the labels?
- Does it reflect elements of the game's design?
- Does it reflect strategies recognized by the game's community?
- How does this information compare to statistical data?
- If you were a developer working on this game, how would you use this information?

In conclusion, in this lab we went through the use of *Stratmapper* to label strategies and behaviors. We used Dota2 data and labeled matches that can be loaded through the *Stratmapper* interface. Further, you were able to gain an understanding of *Stratmapper* interface to analyze spatio-temporal data from a game. We hope this allowed you to understand how powerful visualization can be in the analysis process of player behaviors and understanding the player experience.

For information regarding DotA2 please refer to <http://blog.dota2.com/?l=english> and https://dota2.gamepedia.com/Dota_2_Wiki.