discord activity tracker

VIS - project proposal
study programm: master medieninformatik
at the faculty of computer science and engineering
of cologne university of applied sciences (TH Köln)

submitted by:

name:	Tristan Schmele
matrikel-no.:	11117569
github:	`https://github.com/tschmele/VIS-project`

submitted to: Prof. Dr. Florian Niebling

gummersbach, 03.06.2024

Contents

1 overview **1**

2 describing the dataset **2**
2.1 discord data 2
2.2 pre-processing 4
2.3 stream dataset 6

3 usage scenarios and tasks **7**
3.1 S1 - lonely chatter 7
3.2 S2 - curious streamer 7

4 visualization and sketches **8**
4.1 the timeline 8
4.2 server streamgraph 8
4.3 channel chart 9
4.4 settings segment 9
4.5 migration matrix 9

5 work breakdown and schedule **11**
5.1 milestones 11
5.2 schedule 11

1 overview

viewer-count and viewer-activity are important metrics to evaluate a streamers performance. most streaming platforms provide the streamer with statistics from their own website. however many streamers additionally use separate platforms like discord to have an environment for their community. the activity in those community discord servers is neither included in the statistics of their streaming platform nor is it tracked anywhere usually. to be able to explore the activity within such a discord server and see its relation to streaming activity, i propose building a data visualization that allows streamers to see when and where their server-members have posted messages. this app will show the activity within each text-channel as well as the entire discord server over time, while highlighting the times in which the streamer was live on twitch. it will allow users to explore this dataset by changing the displayed timeframe aswell as filtering on different variables in order to compare user activity under various conditions.

2 describing the dataset

a dataset of discord messages originating from the „Joseph Anderson Discord Server“ (from here as JADS). from this server roughly 40 text-channels and threads have been exported into JSON files.

2.1 discord data

each individual file contains the message history of one channel or thread and follows the same general structure.

```
{
    // specifies which server the current channel originates from
    // largely irrelevant for the scope of this project
    "guild": {
        "id": "",
        "name": "",
        "iconUrl": ""
    },
    // specifies the channels details
    "channel": {
        "id": "",          // [categorical, 42 levels]
        "type": "",        // [categorical, 2 levels]
        "categoryId": "",  // [categorical, 5 levels]
        "category": "",    // [categorical, 5 levels]
        "name": "",        // [categorical, 42 levels]
        "topic": ""        //
    },
    // created by message exporter . potentially exists to export
    // a specific timeframe of messages .
    // always null for all channels . irrelevant for this project
    "dateRange": {
        "after": null,
        "before": null
    },
    // specifies the time at which this channel was exported
    "exportedAt": "",     // [ordered, june 01, 2024 - june 09, 2024]
    // an array containing the individual messages of this channel
    "messages": [],
    // the number of messages in this channel
    "messageCount": 0     // [quantitative, 0 - n]
}
```

Listing 1 discord channel data overview

the "messages" array contains multiple messages which each follow the same pattern as well.

```
"messages": [
    {
        "id": "",                // [categorical, millions levels]
        "type": "",              // [categorical, 32 levels]
        "timestamp": "",  // [ordered, january 01, 2024 - exportedAt]
        "timestampEdited": "",   // [ordered, timestamp - exportedAt]
        "callEndedTimestamp": null,
        "isPinned": false,       // [categorical, binary]
        "content": "",
        // specifies who wrote the message
        "author": {
            "id": "",            // [categorical, 20k levels]
            "name": "",          // [categorical, 20k levels]
            "discriminator": "", // always "0000" , likely a remnant
            "nickname": "",
            "color": "",         // [ordered, "#000000" - "#FFFFFF"]
            "isBot": false,      // [categorical, binary]
            "roles": [],
            "avatarUrl": ""
        },
        // non-text message contents
        "attachements": [],
        "embeds": [],
        "stickers": [],
        // array of reactions attached to this message
        "reactions": [
            {
                // specifies which emoji is used for the reaction
                "emoji": {
                    "id": "",   // [categorical, thousands levels]
                    "name": "", // [categorical, thousands levels]
                    "code": "", // [categorical, thousands levels]
                    "isAnimated": false,  // [categorical, binary]
                    "imageUrl": ""
                },
                "count": 0,    // [quantitative, 1 - n]
```

Listing 2 discord message data overview

```
                // array of users who reacted using this emoji
                "users": [
                    {
                        "id": "",   // [categorical, 20k levels]
                        "name": "", // [categorical, 20k levels]
                        "discriminator": "",
                        "nickname": "",
                        "isBot": false, // [categorical, binary]
                        "avatarUrl": ""
                    }
                ]
            }
        ],
        // array of users mentioned in this message
        "mentions": []
    }
]
```

Listing 3 discord message data continued

2.2 pre-processing

this dataset contains a lot of information that provides little value within the scope of this project. to reduce the size of the overall dataset, the individual files will be trimmed to only contain required information and potentially merged. since this is not a live dataset, this process will be handled in a separate tool and only the trimmed dataset will be fed into the actual pipeline. the original raw data will be stored separately to facilitate adjusting which information gets trimmed or not down the line. the resulting dataset should be formatted like listing 4.

as the entire „guild“ attribute is identical for all channels in this dataset, it has no influence on any analysis and has been removed entirely. from „channel“ most attributes have been kept to help identifying each channel. both „categoryId“ and „category“ have been removed as they serve no purpose with this dataset. most of the included channels have been archived as of exporting this dataset so their current category does not reflect the category they were in while they were active. „topic“ has been kept to be able to display it at a later point as general purpose information about each channel. „dateRange“ has been removed as it is always null for all channels. „exportedAt“ remains untouched to have quick access to the absolute latest possible timestamp within each channel. „messageCount“ remains for a similar purpose. each „message“ has been stripped of most of their content and only contains identifying information as well as the reactions. since the purpose of this project is to work with general activity within the server, the actual content of each message does not provide any benefit as of now. the relevant information is when a message has been posted („timestamp“), who posted the message („author“) and how many users interacted with the message („reactions“). although „reactions“ are a potentially risky metric of activity as they do not contain a timestamp, meaning they could have been added at any time between the creation of a message and the export of this dataset.

```
{
    "channel": {
        "id": "",                   // [categorical, 42 levels]
        "type": "",                 // [categorical, 2 levels]
        "name": "",                 // [categorical, 42 levels]
        "topic": ""                 //
    },
    "exportedAt": "",       // [ordered, june 01,2024 - june 09,2024]
    "messages": [
        {
            "id": "",               // [categorical, millions levels]
            "type": "",             // [categorical, 32 levels]
            "timestamp": "",// [ordered, january 01,2020 - exportedAt]
            "author": {
                "id": "",           // [categorical, 20k levels]
                "name": "",         // [categorical, 20k levels]
                "nickname": "",
                "isBot": false,     // [categorical, binary]
                "avatarUrl": ""
            },
            "reactions": [
                {
                    "emoji": {
                        "id": "",  // [categorical, thousands levels]
                        "name": "",// [categorical, thousands levels]
                        "code": "" // [categorical, thousands levels]
                    },
                    "count": 0,    // [quantitative, 1 - n]
                    "users": [
                        {
                            "id": "",   // [categorical, 20k levels]
                            "isBot": false  // [categorical, binary]
                        }
                    ]
                },
                // ...
            ]
        },
        // ...
    ],
    "messageCount": 0               // [quantitative, 0 - m]
}
```

Listing 4 adjusted discord history dataset structure

2.3 stream dataset

streaming activity will be documented in a separate dataset to facilitate cross-referencing the activity on discord with streams on twitch. the general structure of this dataset is as follows.

```
{
    "games": [
        {
            "id": "",               // [categorical, hundred levels]
            "name": "",             // [categorical, hundred levels]
            "streams": [
                {
                    "id": "",       // [categorical, 749 levels]
                    "start": "",
                          // [ordered, january 01,2020 - exportedAt]
                    "duration": ""  // [quantitative, 0 - n]
                },
                // ...
            ],
            "streamCount": 0        // [quantitative, 0 - 749]
        },
        // ...
    ],
    "gameCount": 0                  // [quantitative, 0 - m]
}
```

Listing 5 structure for the game and stream dataset

3 usage scenarios and tasks

3.1 S1 - lonely chatter

Finn is an australian computer science student. he spends most of his day in university and only has time to chat or play with his friends in the late afternoon, before he goes to bed around midnight to be ready for classes the next day. he repeatedly hears about how hectic the server is, while it is usually calm when he is online. now Finn wants to be able to [explore] the activity in this server, to [compare] the activity at different times and in different channels to [identify] if he is just awake at the wrong times or if he should check out other channels more often.

when Finn opens the „discord activity tracker“ app, he will find an overview of the server activity from the last couple of years split up by channels next to a specific view highlighting only one channel. he can select his most active channel „cringe-cafe“ to be highlighted in the second diagram to see if the activity of this channel is just generally lower than other channels. with the timeline at the top of his screen he can narrow down the displayed timeframe to just the last year which is when he joined the server. he notices that the activity in his favourite channel has dropped off considerably since he first joined the server, even though the activity of the entire server has constantly increased. after looking at the activity over time in multiple other channels he realizes that the channel „dragons-den“ has been consistently more active in the last year when compared to the year before that.

3.2 S2 - curious streamer

Kate has recently started streaming on twitch and has heard of other streamers using discord as a gathering spot for their viewers. she wants to figure out what the implications are of opening new chatrooms for her community outside of twitch. she wants to [explore] a dataset that allows her to [observe] the changes in activity on streamers discord server based on their streaming activity. when Kate opens the „discord activity tracker“ app, she will find an overview of the server activity from the last couple of years split up by channels next to a specific view highlighting one of the channels. she notices right away that the overall activity in the server has been rising over the last years, implying a growth in the community. she can select one or multiple games to highlight the days during which those games have been streamed in the various diagrams. she spots her favourite game „persona 4“ and selects it to be highlighted. she sees that the time from february to april in 2021 has been highlighted in a different colour. during that time the total activity is noticeably increasing. using the calendar icon next to the until now empty „stream migration“ chart she can select one of the days where "persona 4" was streamed. now she can also see which channels the most active users have primarily used during and two hours before and after the stream. after increasing the number of displayed users she notices a trend towards the channels „dragons-den“ and „persona-4-spoiler-chat“. she concludes that using discord in parallel with her streaming activity would be beneficial for her community and decides to provide a dedicated channel to accompany her streams.

4 visualization and sketches

the app will be made up of a full timeline, three main charts and a segment for settings and for selecting streams. two of the charts work together to show the activity of all channels in one while the other displays just one channel that gets highlighted in the first. tool-tips and other changes affect both charts simultaneously.

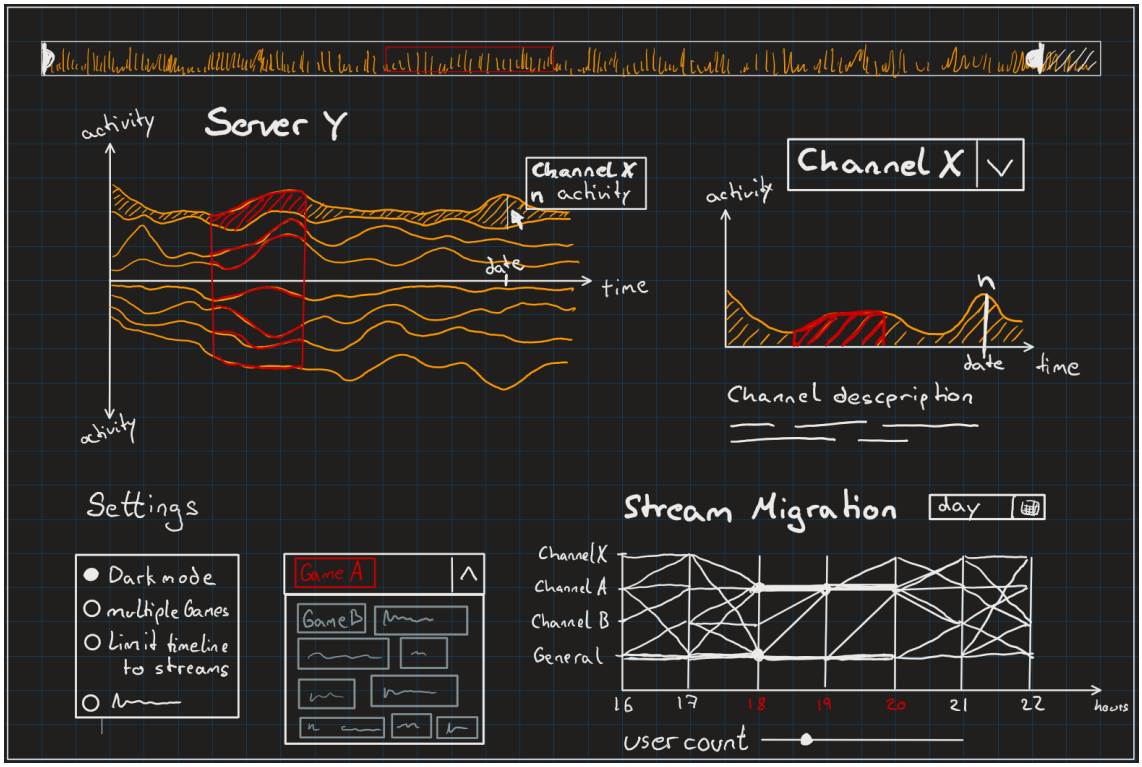

Figure 1 full view of the „discord activity tracker“ app

4.1 the timeline

across the whole width of the screen will be a miniature histogram of the entire timeframe the dataset encompasses. it will serve primarily to visualize the full timeline and to select a specific timeframe for the other views to focus on. limiting the timeframe should work by dragging the ends of the timeline together to mark the start and end of the desired period. alternatively the options should include a function to specifically select the period of time between the first and last stream of selected games.

4.2 server streamgraph

this component displays the server activity within the selected timeframe in a streamgraph. the different layers represent the activity in different channels. hovering within any of the layers shows a tool-tip explaning which channel it represents as well as both its and the entire servers activity at the point in time where the cursor is at.

additionally this view component will have multiple interactions with the channel chart. triggering the tool-tip in either one of those should also show the corresponding tool-tip in the other. if the user clicks on one of the layers, this layer will be highlighted and the content

in the channel chart will change to the selected channel. in reverse if a different channel is selected in the channel chart it will highlight that channel in the server streamgraph.

4.3 channel chart

the channel chart component will show the activity of one selected channel within the timeframe from the timeline. hovering over the chart will display the activity at the corresponding point in time. under this chart will be a field to display the channel description to give a rough idea of what this channel is about. the title of this chart will double as a drop-down menu to select a different channel. the channel chart will either be an area graph or a bar chart.

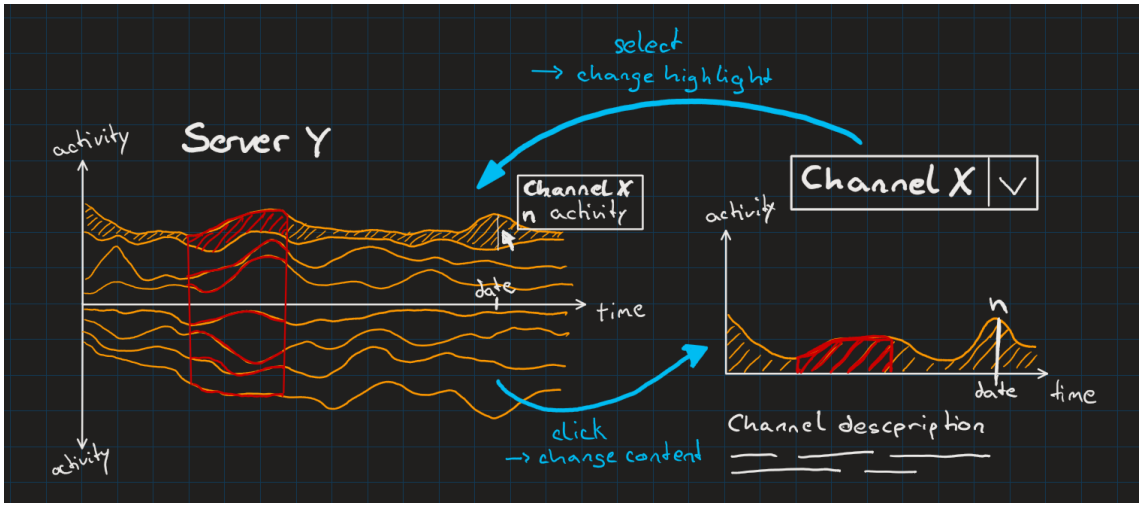

Figure 2 interactivity between the server streamgraph and the channel chart

4.4 settings segment

the settings component will handle potential global settings which could among other things include changing the color-scheme of the various views. most of these settings are not required for the app to work, but can be considered as potential additions later on.

the essential element of this component is the selection of one or more games. for this purpose the element will contain a list of games that have been streamed in the given time period and selecting one or more of them will highlight the period in which these games have been streamed in the server streamgraph, the channel chart and the timeline. this will be combined with a button to automatically adjust the timeline to only select the period from the earliest to the latest stream of the selected games.

4.5 migration matrix

this is going to be the innovative view component. the goal of the migration matrix is to visualize the movement of users between different channels during and around the streams. its a line graph with time on the x-axis and available channels on the y-axis. each line represents one user. the time-axis for this graph is not continuous but only counts in full hour increments and the corresponding point will be put at the respective height of the

channel the user has had the most activity in during the past hour. it will always include the full duration of one stream and two hours before and after the stream.

this view will be accompanied by two interaction components. one is a calendar on which the days with streams of any of the selected games from the settings segment are marked and selecting one of them will change the migration matrix to display the data around that stream. by default it will select the first available day as long as a game has been selected. the other component is a slider to adjust how many users are displayed in the chart. it will sort the users by activity during the available timeframe and start with the 10 most active users, adding users in decreasing activity as the slider is increased.

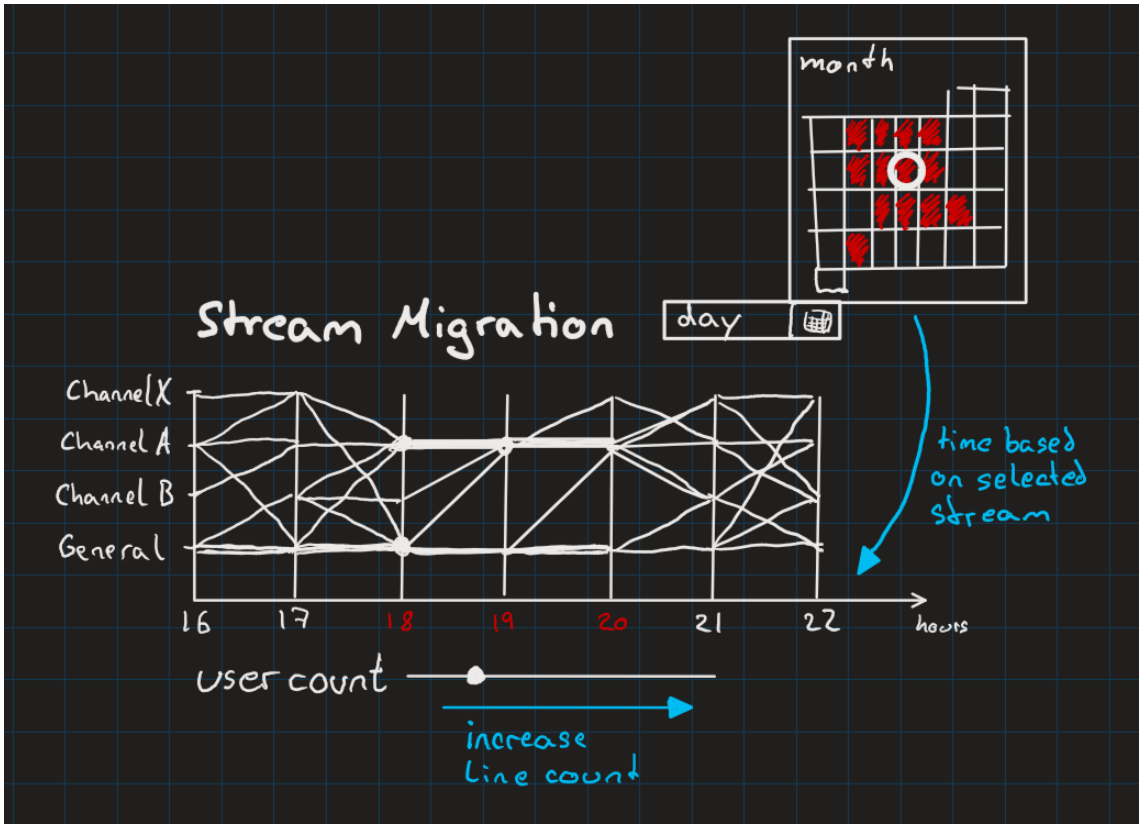

Figure 3 close-up of the migration view with further explanation and calendar component

5 work breakdown and schedule

the work will be broken down into milestones (M1 etc) and tasks (T1-1 etc). each milestone will be made up of multiple tasks that should be complete at this milestone.

5.1 milestones

M1 dataset pre-processing
- T1-1 download all required discord chat histories
- T1-2 create the stream dataset
- T1-3 adjust the discord history dataset

M2 static activity views
- T2-1 implement activity calculation
- T2-2 implement static channel chart
- T2-3 implement static streamgraph
- T2-4 implement timeline histogram

M3 activity interactivity
- T3-1 implement tool-tips
- T3-2 implement streamgraph and channel chart updating
- T3-3 implement timeline adjustment

M4 stream highlighting
- T4-1 implement selection of games
- T4-2 add game-based highlighting to existing components
- T4-3 implement game-based timeline selection

M5 migration matrix
- T5-1 implement most active channel calculation
- T5-2 implement static line chart
- T5-3 add date-based updating
- T5-4 add user count slider updating

5.2 schedule

M1 - 15 hours - 23.06.2024

M2 - 18 hours - 07.07.2024

M3 - 22 hours - 28.07.2024

M4 - 10 hours - 11.08.2024

M5 - 25 hours - 08.09.2024